

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

**ESCUELA PROFESIONAL DE INGENIERÍA EN
COMPUTACIÓN E INFORMÁTICA**



TESIS

**“SISTEMA DE GESTIÓN, ANÁLISIS Y CARTOGRAFÍA PARA
EL SEGURO AGRÍCOLA CATASTRÓFICO, LIMA - 2017”**

INVESTIGADORES:

Bach. RODAS ESCURRA CHRISTIAN CESAR

Bach. AMAYA PEREYRA CINTHYA VIVIANA

ASESOR:

Ing. MARTÍN LEIVA CASTILLO

LAMBAYEQUE, 2019



**UNIVERSIDAD NACIONAL
PEDRO RUIZ GALLO**



**FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA EN
COMPUTACIÓN E INFORMÁTICA**

**“SISTEMA DE GESTIÓN, ANÁLISIS Y CARTOGRAFÍA PARA EL
SEGURO AGRÍCOLA CATASTRÓFICO LIMA - 2017”**

Presentado para optar el Título Profesional de:

Ingeniero en Computación e Informática

Presentado por:

**Bach. Rodas Ecurra Christian Cesar
INVESTIGADOR**

**Bach. Amaya Pereyra Cinthya Viviana
INVESTIGADORA**

**Ing. Martín Leiva Castillo
ASESOR**



**UNIVERSIDAD NACIONAL
PEDRO RUIZ GALLO**



**FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA EN
COMPUTACIÓN E INFORMÁTICA**

**“SISTEMA DE GESTIÓN, ANÁLISIS Y CARTOGRAFÍA PARA EL
SEGURO AGRÍCOLA CATASTRÓFICO LIMA - 2017”**

Presentado para optar el Título Profesional de:

Ingeniero en Computación e Informática

Aprobado por:

Dr. Ing. Armando José Moreno Heredia
PRESIDENTE

M.Sc. Ing. Roger Ernesto Alarcón García
SECRETARIO

Dra. Ing. Giuliana Fiorella Lecca Orrego
VOCAL

ÍNDICE

AGRADECIMIENTO	xiv
RESUMEN	xv
INTRODUCCIÓN	1
CAPÍTULO I: DISEÑO TEÓRICO.....	2
1. REALIDAD PROBLEMÁTICA	2
<u>A.</u> SITUACIÓN PROBLEMÁTICA INTERNACIONAL	2
<u>B.</u> SITUACIÓN PROBLEMÁTICA NACIONAL- LOCAL	4
2. FORMULACIÓN DEL PROBLEMA	5
3. OBJETO DE ESTUDIO Y CAMPOS DE ACCIÓN.....	5
4. DELIMITACIÓN DE LA INVESTIGACIÓN	6
5. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN	6
<u>A.</u> ASPECTO TECNOLÓGICO	6
<u>B.</u> ASPECTO ECONÓMICO	7
<u>I.</u> COSTOS DIRECTOS	7
<u>II.</u> COSTOS INDIRECTOS	7
<u>III.</u> ASPECTO SOCIAL	7
6. LIMITACIONES DE LA INVESTIGACIÓN.....	8
7. OBJETIVOS DE LA INVESTIGACIÓN.....	8
<u>A.</u> OBJETIVO GENERAL	8
<u>B.</u> OBJETIVO ESPECÍFICO	8
8. ASPECTO DE LA ORGANIZACIÓN.....	9
<u>A.</u> NATURALEZA.....	9
<u>I.</u> MINISTERIO DE AGRICULTURA Y RIEGO	9
<u>II.</u> DIRECCIÓN REGIONAL DE AGRICULTURA	10
<u>III.</u> LA POSITIVA SEGUROS Y REASEGUROS.....	10
<u>IV.</u> AGROEVALUACIONES PERÚ SRL.....	11
<u>B.</u> JURISDICCIÓN	11
<u>I.</u> MINISTERIO DE AGRICULTURA Y RIEGO	11
<u>II.</u> DIRECCIÓN REGIONAL AGRARIA	13
<u>III.</u> LA POSITIVA SEGUROS Y REASEGUROS.....	13
<u>IV.</u> AGROEVALUACIONES PERÚ SRL.....	13
<u>C.</u> DOMICILIO LEGAL	13
<u>I.</u> MINISTERIO DE AGRICULTURA Y RIEGO	13
<u>II.</u> DIRECCIÓN REGIONAL AGRARIA	14
<u>III.</u> LA POSITIVA SEGUROS Y REASEGUROS.....	14
<u>IV.</u> AGROEVALUACIONES PERÚ SRL.....	14
<u>D.</u> DESCRIPCIÓN.....	14
<u>I.</u> MINISTERIO DE AGRICULTURA Y RIEGO	14
<u>II.</u> DIRECCIÓN REGIONAL DE AGRICULTURA	15
<u>III.</u> LA POSITIVA SEGUROS Y REASEGUROS.....	16
<u>IV.</u> AGROEVALUACIONES PERÚ SRL.....	16
<u>E.</u> ORGANIGRAMA	17
<u>I.</u> MINISTERIO DE AGRICULTURA Y RIEGO	17
<u>II.</u> DIRECCIÓN REGIONAL AGRARIA	18
<u>III.</u> LA POSITIVA SEGUROS Y REASEGUROS.....	19
<u>IV.</u> AGROEVALUACIONES PERÚ SRL.....	20
<u>F.</u> MISIÓN Y VISIÓN	20
<u>I.</u> MISIÓN	20
<u>1.</u> MINISTERIO DE AGRICULTURA Y RIEGO	20
<u>2.</u> DIRECCIÓN REGIONAL AGRARIA	20
<u>3.</u> LA POSITIVA SEGUROS Y REASEGUROS.....	20
<u>4.</u> AGROEVALUACIONES PERÚ SRL.....	21

<u>II.</u>	VISIÓN.....	21
<u>1.</u>	MINISTERIO DE AGRICULTURA Y RIEGO.....	21
<u>2.</u>	DIRECCIÓN REGIONAL AGRARIA	21
<u>3.</u>	LA POSITIVA SEGUROS Y REASEGUROS	21
<u>4.</u>	AGROEVALUACIONES PERÚ SRL.....	21
9.	ANTECEDENTES DE ESTUDIO	22
<u>A.</u>	A NIVEL INTERNACIONAL	22
<u>B.</u>	A NIVEL NACIONAL	23
<u>C.</u>	A NIVEL LOCAL.....	24
	CAPÍTULO II: MÉTODOS Y MATERIALES.....	25
1.	BASE TEÓRICA	25
<u>A.</u>	SISTEMAS DE INFORMACIÓN GEOGRÁFICA SIG	25
2.	ENFOQUES ÁGILES	25
<u>A.</u>	MANIFIESTO ÁGIL	25
<u>B.</u>	PRINCIPIOS DEL MANIFIESTO ÁGIL:.....	27
3.	METODOLOGÍA POR USAR.....	28
<u>A.</u>	ANÁLISIS DE LA METODOLOGÍA POR USAR	28
<u>B.</u>	PROGRAMACIÓN EXTREMA (XP).....	30
<u>I.</u>	FASES DE LA METODOLOGÍA XP	30
<u>II.</u>	LAS HISTORIAS DE USUARIO XP	32
<u>III.</u>	ROLES XP	33
<u>IV.</u>	PRÁCTICAS XP	34
4.	HERRAMIENTAS	39
<u>A.</u>	SISTEMA WEB.....	39
<u>I.</u>	INTERFAZ WEB	39
<u>1.</u>	PYTHON.....	40
<u>2.</u>	DJANGO:	42
<u>3.</u>	BOOT STRAP.....	44
<u>4.</u>	JAVASCRIPT	45
<u>5.</u>	JQUERY.....	46
<u>6.</u>	HTML.....	46
<u>7.</u>	CSS	47
<u>II.</u>	SISTEMA OPERATIVO.....	48
<u>1.</u>	MYSQL	48
<u>2.</u>	NGINX	49
<u>3.</u>	UWSGI.....	50
<u>4.</u>	VIRTUAL ENV	50
<u>5.</u>	GIT	51
5.	DESCRIPCIÓN GRÁFICA DE CÓMO TRABAJAN ESTAS HERRAMIENTAS	53
	CAPÍTULO III: RESULTADOS Y DISCUSIÓN.....	54
1.	METODOLOGÍA XP.....	54
<u>I.</u>	PLANIFICACIÓN DEL PROYECTO.....	54
<u>A.</u>	REQUERIMIENTOS FUNCIONALES:	54
<u>I.</u>	ITERACIÓN I.....	54
<u>II.</u>	ITERACIÓN II	55
<u>III.</u>	ITERACIÓN III.....	62
<u>IV.</u>	ITERACIÓN IV.....	67
<u>V.</u>	ITERACIÓN V	71
<u>VI.</u>	ITERACIÓN VI.....	76
<u>VII.</u>	ITERACIÓN VII	81
<u>VIII.</u>	ITERACIÓN VIII	84
<u>B.</u>	REQUERIMIENTOS NO FUNCIONALES	88
<u>II.</u>	DISEÑO DEL PROYECTO.....	89
<u>A.</u>	DISEÑO LÓGICO	89
<u>B.</u>	MODELO DEL NEGOCIO	91
<u>C.</u>	DIAGRAMA DE ESTADOS.....	92

<u>D.</u>	MODELO DE BASE DE DATOS.....	93
<u>E.</u>	DIAGRAMA DE COMPONENTES	94
<u>F.</u>	DIAGRAMA DE DESPLIEGUE	95
A.	DICCIONARIO DE DATOS.....	96
B.	TARJETAS CRC	106
<u>III.</u>	CODIFICACIÓN DEL PROYECTO	110
<u>A.</u>	ITERACIÓN I.....	110
<u>B.</u>	ITERACIÓN II.....	112
<u>C.</u>	ITERACIÓN III	121
<u>D.</u>	ITERACIÓN IV	125
<u>E.</u>	ITERACIÓN IV	128
<u>F.</u>	ITERACIÓN V	132
<u>G.</u>	ITERACIÓN VII.....	136
<u>H.</u>	ITERACIÓN VIII	139
<u>IV.</u>	PRUEBAS UNITARIAS DEL PROYECTO Y DE INTEGRACIÓN	142
<u>V.</u>	RESULTADOS:	154
	CAPÍTULO IV: CONCLUSIONES	155
	CAPÍTULO V: RECOMENDACIONES	156
	BIBLIOGRAFÍA REFERENCIADA	157
	TRABAJOS CITADOS	158
<u>I.</u>	ANEXO 1: ENTENDIMIENTO DEL NEGOCIO:.....	159
A.	JUSTIFICACIÓN DEL MACROPROCESO BAJO ESTUDIO	160
✓	DEFINAMOS LA INTERFAZ DE NUESTRO SISTEMA:	160
✓	DEFINAMOS LOS ESTADOS.....	160
✓	ESTADO INGRESADO:	160
✓	ESTADO REVISADO:	161
✓	ESTADO AGREGADO:	163
✓	ESTADO ASIGNADO:.....	164
✓	ESTADO EVALUADO:	165
✓	ESTADO TERMINADO:	169
<u>III.</u>	ANEXO 2: FACTIBILIDAD DEL PROYECTO:.....	170
<u>IV.</u>	ANEXO 3: CRONOGRAMA DE ACTIVIDADES:	176

TABLAS

TABLA 1:	COMPARATIVO DE METODOLOGÍAS ÁGILES	29
TABLA 2:	HU N°1: INTEGRAR CARTOGRAFÍA CON SECTORES ESTADÍSTICOS	54
TABLA 3:	HU N°2: REGISTRAR AVISO	56
TABLA 4:	HU N°3: GENERAR MAPA DEL SECTOR ESTADÍSTICO CON PUNTOS A EVALUAR – WEB	57
TABLA 5:	HU N° 4: MOSTRAR MAPA DEL SECTOR ESTADÍSTICO – TABLET	59
TABLA 6:	HU N° 5: ADMINISTRAR BANDEJA DE AVISOS	60
TABLA 7:	HU N° 6: DESCARGAR AVISOS EN TABLET	61
TABLA 8:	HU N°7: REGISTRAR DATOS DE SINIESTROS POR AVISO – TABLET	63
TABLA 9:	HU N°8: ENVIAR AVISOS TERMINADOS - TABLET:	64
TABLA 10:	HU N°9: AVISOS ENVIADOS.....	65
TABLA 11:	HU N ° 10: CERRAR EL AVISO – WEB.....	67
TABLA 12:	HU N°11: ENVIAR AVISO A POSITIVA	68
TABLA 13:	HU N ° 12: ENVIAR AVISO AL AJUSTADOR.....	69
TABLA 14:	HU N °13: ENVIAR AVISO AL EVALUADOR	70
TABLA 15:	HU N °14: RETORNAR AVISOS AL AJUSTADOR	71
TABLA 16:	HU N°15: RETORNAR AVISOS A POSITIVA.....	72
TABLA 17:	HU N°16: RETORNAR AVISOS A MINAGRI	73
TABLA 18:	HU N°17: RETORNAR AVISOS AL REPRESENTANTE DE REGIÓN	75

TABLA 19: HU N°18: ASIGNAR EVALUADOR	76
TABLA 20: HU N°19: PROGRAMAR EVALUACIÓN	78
TABLA 21: HU N°20: AGREGAR DATOS A AVISO	79
TABLA 22: HU N° 22: EVALUAR AVISOS VÍA WEB	80
TABLA 23: HU N° 22: INICIO DE SESIÓN	81
TABLA 24: HU N°23: BUSCAR AVISO	82
TABLA 25: HU N°24: EXPORTAR AVISOS EN EXCEL	83
TABLA 26: HU N° 25: DESCARGAR ACTAS DE ATENCIÓN POR AVISO	85
TABLA 27: HU N °26: VER ESTADO DE AVISO	86
TABLA 28: HU N° 27: LISTAR AVISOS	87
TABLA 29: REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA	88
TABLA 30: ROLES DEL APLICATIVO	96
TABLA 31: PERMISOS DE LOS GRUPOS EN EL APLICATIVO	96
TABLA 32: ACCIONES PERMITIDAS EN EL SISTEMA	97
TABLA 33: USUARIOS DE LOS APLICATIVOS	97
TABLA 34: RELACIÓN ENTRE GRUPOS Y USUARIOS	98
TABLA 35: PERMISOS A LAS ACCIONES DE LOS USUARIOS	98
TABLA 36: LOG DE ACCIONES -ADMINISTRADOR DJANGO	99
TABLA 37: MODELOS DE TABLAS	99
TABLA 38: CONTROL DE MIGRACIONES	100
TABLA 39: REGISTRO DE INICIO DE SESIÓN	100
TABLA 40: DATOS DE LOS AVISOS INGRESADOS	101
TABLA 41: COMENTARIOS DE LOS USUARIOS EN LOS AVISOS	102
TABLA 42: TABLA DE DEPARTAMENTOS	102
TABLA 43: TABLA DE PROVINCIAS	103
TABLA 44: TABLA DE DISTRITOS	103
TABLA 45: TABLA DE SECTOR ESTADÍSTICO	104
TABLA 46: CONTROL DE LOGINS	104
TABLA 47: GENERAR PUNTOS Y MUESTRA DE EVALUACIÓN DE LOS AVISOS	105
TABLA 48: CRC AUTH_USER	106
TABLA 49: CRC AUTH_GROUP	106
TABLA 50: CRC LOGLOGIN	106
TABLA 51: CRC DEPARTAMENTO	107
TABLA 52: CRC PROVINCIA	107
TABLA 53: CRC DISTRITO	107
TABLA 54: CRC SECTOR ESTADÍSTICO	108
TABLA 55: CRC CULTIVO	108
TABLA 56: CRC AVISO	108
TABLA 57: CRC PUNTO DE EVALUACIÓN	109
TABLA 58: CRC REGISTRAR CAMBIOS DE ESTADO DE LOS AVISOS	109
TABLA 59: CRC REGISTRAR COMENTARIOS DE LOS AVISOS	109
TABLA 60: ANÁLISIS DE PRODUCTIVIDAD DEL SISTEMA	154
TABLA 61: HISTORIAL DE PLANIFICACIÓN Y DESARROLLO DEL SISTEMA	159
TABLA 62: FLUJO DE CAJA ACTUAL – AGROEVALUACIONES	171
TABLA 63: FLUJO DE CAJA PROYECTADO – AGROEVALUACIONES	172
TABLA 64: RELACIÓN COSTO- BENEFICIO, VAN Y TIR	173
TABLA 65: RESUMEN DE EVALUACIÓN ECONÓMICA - VAN-TIR-B/C	174

IMÁGENES

IMAGEN 1: ORGANIGRAMA DE MINISTERIO DE AGRICULTURA Y RIEGO (MINAGRI)	17
IMAGEN 2: ORGANIGRAMA DIRECCIÓN REGIONAL AGRARIA.....	18
IMAGEN 3: ORGANIGRAMA LA POSITIVA SEGUROS	19
IMAGEN 4: ORGANIGRAMA AGROEVALUACIONES PERÚ SRL	20
IMAGEN 5: ARQUITECTURA A NIVEL DE SOFTWARE DEL SISTEMA DE GESTIÓN, ANÁLISIS Y CARTOGRAFÍA PARA EL SEGURO AGRÍCOLA CATASTRÓFICO LIMA – 2017.....	53
IMAGEN 6: PROTOTIPO INTEGRAR CARTOGRAFÍA CON SECTORES ESTADÍSTICOS.....	55
IMAGEN 7: PROTOTIPO REGISTRAR AVISO	56
IMAGEN 8: PROTOTIPO GENERAR MAPA DEL SECTOR ESTADÍSTICO CON PUNTOS A EVALUAR – WEB	58
IMAGEN 9: PROTOTIPO MOSTRAR MAPA DEL SECTOR ESTADÍSTICO – TABLET.....	59
IMAGEN 10: PROTOTIPO ADMINISTRAR BANDEJA DE AVISOS	60
IMAGEN 11: PROTOTIPO DESCARGAR AVISOS EN TABLET	61
IMAGEN 12: PROTOTIPO REGISTRAR SINIESTROS POR AVISO – TABLET	63
IMAGEN 13:PROTOTIPO ENVIAR AVISOS TERMINADOS – TABLET.....	65
IMAGEN 14: PROTOTIPO AVISOS ENVIADOS	66
IMAGEN 15: PROTOTIPO CERRAR EL AVISO – WEB	67
IMAGEN 16: PROTOTIPO ENVIAR AVISO A POSITIVA	68
IMAGEN 17: PROTOTIPO ENVIAR AVISOS AL AJUSTADOR	69
IMAGEN 18: PROTOTIPO ENVIAR AVISO AL AJUSTADOR	70
IMAGEN 19: PROTOTIPO RETORNAR AVISO AL AJUSTADOR	72
IMAGEN 20: PROTOTIPO: RETORNAR AVISO A POSITIVA	73
IMAGEN 21:PROTOTIPO REGRESAR AVISOS AL MINAGRI	74
IMAGEN 22: PROTOTIPO: RETORNAR AVISOS AL REPRESENTANTE DE REGIÓN	75
IMAGEN 23: PROTOTIPO ASIGNAR EVALUADOR.....	77
IMAGEN 24: PROTOTIPO PROGRAMAR EVALUACIÓN	78
IMAGEN 25: PROTOCOLO AGREGAR DATOS A AVISO.....	79
IMAGEN 26:PROTOTIPO EVALUAR AVISOS VÍA WEB	80
IMAGEN 27: PROTOTIPO INICIO DE SESIÓN.....	81
IMAGEN 28: PROTOTIPO: BUSCAR AVISO	82
IMAGEN 29: PROTOTIPO EXPORTAR A EXCEL	84
IMAGEN 30: PROTOTIPO DESCARGAR ACTAS DE ATENCIÓN POR AVISO.....	85
IMAGEN 31: PROTOTIPO VER ESTADO DE LOS AVISOS	86
IMAGEN 32: PROTOTIPO LISTAR AVISO	87
IMAGEN 33: DIAGRAMA LÓGICO 1 ERA PARTE.....	89
IMAGEN 34: DIAGRAMA LÓGICO 2 DA PARTE.....	90
IMAGEN 35 : MODELO DE NEGOCIO.....	91
IMAGEN 36: DIAGRAMA DE ESTADOS	92
IMAGEN 37: MODELO DE BASE DE DATOS	93
IMAGEN 38: DIAGRAMA DE COMPONENTES	94
IMAGEN 39: DIAGRAMA DE DESPLIEGUE.....	95
IMAGEN 40: CODIFICACIÓN MODELOS/ TABLAS A USAR EN LA HU N°1.....	110
IMAGEN 41: CODIFICACIÓN DEL PROCESO PRINCIPAL PARA GENERAR EL MAPA DE LA HUN°1	111
IMAGEN 42:RESULTADO DE LA CODIFICACIÓN DE LA HU N°1.....	111
IMAGEN 43: CODIFICACIÓN MODELOS/ TABLAS A USAR EN LA HU N°2.....	112
IMAGEN 44: CODIFICACIÓN DEL PROCESO PRINCIPAL PARA REGISTRAR EL AVISO DE LA HUN°2.....	113
IMAGEN 45:RESULTADO DE LA CODIFICACIÓN DE LA HU N°2	113
IMAGEN 46: CODIFICACIÓN MODELOS/ TABLAS A USAR EN LA HU N°3.....	114

IMAGEN 47:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA GENERAR MAPA Y PUNTOS EN LA WEB DE LA HUN°3	115
IMAGEN 48: RESULTADO DE LA CODIFICACIÓN DE LA HU N°3.....	115
IMAGEN 49:CODIFICACIÓN MODELOS/ TABLAS A USAR EN LA HU N°4.....	116
IMAGEN 50:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA ENVÍA URL DE MAPAS DESDE LA WEB HASTA LAS TABLETS DE LA HUN°4	117
IMAGEN 51: PROCESO PRINCIPAL PARA DESCARGAR LA URL DEL MAPA EN LA TABLET DE LA HUN°4	117
IMAGEN 52: RESULTADO DE LA CODIFICACIÓN DE LA HU N°4	118
IMAGEN 53:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA ADMINISTRAR BANDEJAS DE AVISOS EN LA WEB DE LA HUN°5	119
IMAGEN 54:RESULTADO DE LA CODIFICACIÓN DE LA HU N°5.....	119
IMAGEN 55:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA DESCARGAR AVISOS EN LA TABLET DE LA HUN°6.....	120
IMAGEN 56:RESULTADO DE LA CODIFICACIÓN DE LA HU N°6	120
IMAGEN 57:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA REGISTRAR DATOS DE SINIESTRO POR AVISO EN LA TABLET DE LA HUN°7	121
IMAGEN 58:RESULTADO DE LA CODIFICACIÓN DE LA HU N°7	121
IMAGEN 59:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA ENVIAR AVISOS TERMINADOS DE LA TABLET AL SERVIDOR DE LA HUN°8	122
IMAGEN 60:RESULTADO DE LA CODIFICACIÓN DE LA HU N°8	122
IMAGEN 61:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA VISUALIZAR LA BANDEJA DE AVISOS ENVIADOS DESDE LA TABLET DE LA HUN°9	123
IMAGEN 62:RESULTADO DE LA CODIFICACIÓN DE LA HU N°9	123
IMAGEN 63:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA CERRAR AVISO DESDE LA WEB DE LA HUN°10	124
IMAGEN 64:RESULTADO DE LA CODIFICACIÓN DE LA HU N°10.....	124
IMAGEN 65:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA ENVIAR AVISOS A POSITIVA DE LA HUN°11	125
IMAGEN 66:RESULTADO DE LA CODIFICACIÓN DE LA HU N°11.....	125
IMAGEN 67:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA ENVIAR LOS AVISOS AL AJUSTADOR DE LA HUN°12	126
IMAGEN 68:RESULTADO DE LA CODIFICACIÓN DE LA HU N°12.....	126
IMAGEN 69:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA ENVIAR LOS AVISOS AL EVALUADOR DE LA HUN°13	127
IMAGEN 70:RESULTADO DE LA CODIFICACIÓN DE LA HU N°13.....	127
IMAGEN 71:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA RETORNAR LOS AVISOS AL AJUSTADOR DE LA HUN°14.....	128
IMAGEN 72:RESULTADO DE LA CODIFICACIÓN DE LA HU N°14.....	128
IMAGEN 73:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA RETORNAR LOS AVISOS A POSITIVA DE LA HUN°15	129
IMAGEN 74:RESULTADO DE LA CODIFICACIÓN DE LA HU N°15.....	129
IMAGEN 75:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA RETORNAR LOS AVISOS A MINAGRI DE LA HUN°16.....	130
IMAGEN 76:RESULTADO DE LA CODIFICACIÓN DE LA HU N°16.....	130
IMAGEN 77:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA RETORNAR LOS AVISOS A AL REPRESENTANTE DE REGIÓN DE LA HUN°17	131
IMAGEN 78:RESULTADO DE LA CODIFICACIÓN DE LA HU N°17.....	131
IMAGEN 79:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA ASIGNAR EVALUADOR DE LA HUN°18.....	132
IMAGEN 80:RESULTADO DE LA CODIFICACIÓN DE LA HU N°18.....	132
IMAGEN 81:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA PROGRAMAR EVALUACIÓN DE LA HUN°19	133
IMAGEN 82: RESULTADO DE LA CODIFICACIÓN DE LA HU N°19.....	133
IMAGEN 83:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA AGREGAR DATOS DE LA HUN°20	134

IMAGEN 84:RESULTADO DE LA CODIFICACIÓN DE LA HU N°20.....	134
IMAGEN 85:DIFICACIÓN DEL PROCESO PRINCIPAL PARA EVALUAR AVISOS VÍA WEB DE LA HUN°21.....	135
IMAGEN 86:RESULTADO DE LA CODIFICACIÓN DE LA HU N°21.....	135
IMAGEN 87:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA EL INICIO DE SESIÓN DE LA HUN°22.....	136
IMAGEN 88:RESULTADO DE LA CODIFICACIÓN DE LA HU N°22.....	136
IMAGEN 89:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA BUSCAR AVISO DE LA HUN°2	137
IMAGEN 90:RESULTADO DE LA CODIFICACIÓN DE LA HU N°23.....	137
IMAGEN 91:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA EXPORTAR AVISOS EN EXCEL DE LA HUN°24	138
IMAGEN 92:RESULTADO DE LA CODIFICACIÓN DE LA HU N°24.....	138
IMAGEN 93:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA DESCARGAR ACTAS DE ATENCIÓN POR AVISO DE LA HUN°25	139
IMAGEN 94:RESULTADO DE LA CODIFICACIÓN DE LA HU N°25.....	139
IMAGEN 95:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA VER ESTADOS DE LOS AVISOS DE LA HUN°26	140
IMAGEN 96:RESULTADO DE LA CODIFICACIÓN DE LA HU N°26.....	140
IMAGEN 97:CODIFICACIÓN DEL PROCESO PRINCIPAL PARA LISTAR LOS AVISOS DE LA HUN°27.....	141
IMAGEN 98:RESULTADO DE LA CODIFICACIÓN DE LA HU N°27.....	141
IMAGEN 99: CONFIGURACIÓN DE TEST - PRUEBAS UNITARIAS.....	142
IMAGEN 100: PRUEBA UNITARIA LOGIN.....	142
IMAGEN 101: PRUEBA UNITARIA REGISTRAR AVISOS.....	143
IMAGEN 102:PRUEBA UNITARIA - ENVÍO DE DATOS DESDE TABLET.....	143
IMAGEN 103: PRUEBA UNITARIA - ENVÍO DE DATOS AL MINAGRI	144
IMAGEN 104: PRUEBA UNITARIA- RETORNO DE INFORMACIÓN AL REPRESENTANTE DE REGIÓN.....	145
IMAGEN 105: PRUEBA UNITARIA - ENVÍO DE DATOS A POSITIVA	146
IMAGEN 106: PRUEBA UNITARIA- RETORNO DE INFORMACIÓN AL MINAGRI.....	147
IMAGEN 107: PRUEBA UNITARIA - ENVÍO DE DATOS AL AJUSTADOR	148
IMAGEN 108: IMAGEN 105: PRUEBA UNITARIA- RETORNO DE INFORMACIÓN A POSITIVA ..	149
IMAGEN 109: PRUEBA UNITARIA - ENVÍO DE DATOS AL EVALUADOR.....	150
IMAGEN 110:PRUEBA UNITARIA- RETORNO DE INFORMACIÓN AL AJUSTADOR	151
IMAGEN 111: PRUEBA UNITARIA - TERMINAR AVISO.....	152
IMAGEN 112: RESULTADO DE LAS PRUEBAS UNITARIAS	153
IMAGEN 113: INTERFAZ DEL SISTEMA	160
IMAGEN 114: INGRESO DE DATOS AL SISTEMA – RR.....	161
IMAGEN 115:BANDEJA DE AVISOS INGRESADOS DEL MINAGRI	162
IMAGEN 116: DETALLE DE AVISO – MINAGRI	162
IMAGEN 117: PUNTOS A EVALUAR – MINAGRI	163
IMAGEN 118: BANDEJA DE AVISOS - LA POSITIVA	163
IMAGEN 119: AGREGAR DATOS - LA POSITIVA.....	164
IMAGEN 120: ASIGNAR DATOS DE AVISO – AJUSTADOR	165
IMAGEN 121: BANDEJA DE AVISOS RECIBIDOS EN TABLET – EVALUADOR.....	166
IMAGEN 122: EVALUAR O DESESTIMAR AVISO EN CAMPO.....	167
IMAGEN 123: MAPA QUE SE MUESTRA EN LA TABLET.....	167
IMAGEN 124: REVISIÓN DE AVISOS EVALUADOS ENVIADOS AL SERVIDOR – EVALUADORES	168
IMAGEN 125: CERRAR EVALUACIÓN – EVALUADORES.....	169
IMAGEN 126: CRONOGRAMA DE ACTIVIDADES	176

DECLARACIÓN JURADA DE ORIGINALIDAD

Nosotros, **Rodas Ecurra Christian Cesar** y **Amaya Pereyra Cinthya Viviana**, Investigadores Principales y el **Ing. Martín Leiva Castillo**, Asesor del Trabajo de Investigación: “Sistema de gestión, análisis y cartografía para el seguro agrícola catastrófico Lima - 2017”, declaramos bajo juramento que este trabajo no ha sido plagiado, ni contiene datos falsos.

En caso se demuestre lo contrario, asumo responsablemente la anulación de este informe y por ende el proceso administrativo a que hubiera lugar.

Que puede conducir a la anulación del título o grado emitido como consecuencia de este informe.

Lambayeque 25 de octubre de 2019;

Bach. Rodas Ecurra Christian Cesar
INVESTIGADOR

Bach. Amaya Pereyra Cinthya Viviana
INVESTIGADORA

Ing. Martín Leiva Castillo
ASESOR

ACTA DE SUSTENTACIÓN



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DECANATO
Ciudad Universitaria - Lambayeque



ACTA DE SUSTENTACIÓN N° 062-2019-D/FACFyM

(Sustentación Autorizada por Resolución N° 1255-2019-D/FACFyM)

En la ciudad de Lambayeque, siendo las 12 m del día 25 de Octubre del 2019 se reunieron en la Videoteca del Laboratorio de Física los miembros del Jurado designados mediante Resolución N° 884-2017-D/FACFyM, los docentes:

Dr. Ing. Armando José Moreno Heredia	Presidente
M.Sc. Ing. Roger Ernesto Alarcón García	Secretario
Dra. Giuliana Fiorella Lecca Orrego	Vocal

Para recibir la tesis titulada:

'Sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico, Lima - 2017'


desarrollada por los Bachilleres en Computación e Informática, **Rodas Escurra Christian César y Amaya Pereyra Cinthya Viviana.**

Después de escuchar la exposición y las respuestas a las preguntas formuladas por los miembros del Jurado, se acordó APROBAR el trabajo por UNANIMIDAD con el calificativo de MUY BUENO.

En consecuencia, los Bachilleres en referencia quedan aptos para recibir el Título Profesional de **Ingeniero en Computación e Informática**, de acuerdo a la Ley Universitaria, el Estatuto y Reglamento de la Universidad Nacional Pedro Ruiz Gallo de Lambayeque.

Observaciones:

Para constancia del hecho firman.


Dr. Ing. Armando José Moreno Heredia
Presidente


M.Sc. Ing. Roger Ernesto Alarcón García
Secretario


Dra. Ing. Giuliana Fiorella Lecca Orrego
Vocal

DEDICATORIA

Este trabajo de investigación se lo dedico a Dios, quien es mi guía y fortaleza en todo momento; a mi querido y amado padre Antonio Alfredo Amaya Ventura, quien me enseñó a no rendirme y luchar hasta el final, a mi amorosa madre Yris Lupita Pereyra Paz, quien me ayuda a ser una mejor versión de mí día a día y a mi novio Christian Cesar Rodas Ecurra, que es la persona que comparte mis alegrías y fracasos durante estos últimos 9 años, siempre motivándome a ser una mejor profesional y compañera de vida, como él dice “Juntos, vamos por mucho más”.

Cinthya Viviana Amaya Pereyra

Dedico este trabajo de investigación a Dios, y a mis padres quien son mi guía y mi fortaleza; a mi novia Cinthya quien es el principal artífice de este trabajo. Asimismo, dedico este trabajo al “Aco” para que le sirva de fuente de inspiración, demostrándole que puede lograr todo lo que se proponga en la vida, entendiendo que no existe límites para sus sueños.

Christian Cesar Rodas Ecurra

AGRADECIMIENTO

A Dios por permitirme cumplir este objetivo, también agradecer a mis padres que siempre estuvieron apoyándome, aconsejándome y por su infinito amor. Agradecer a mis queridos primos Selene Molina, Almendra Molina, Paola Amaya y Juan Pablo Pereyra quienes siempre están apoyándome incondicionalmente, emocionalmente y por la consideración que me tienen, estoy orgullosa y satisfecha con la familia que tocó, por eso muchas gracias a la familia Amaya – Pereyra. No quisiera terminar este agradecimiento sin antes reconocer todo lo que la familia Rodas Ecurra ha aportado para cumplir este objetivo, muchas gracias por todo el cariño, comprensión y atención que tienen conmigo. Y, por último, pero no menos importante, agradecer toda la paciencia de nuestro asesor el Ing. Martín Leiva, quien nos brindó su tiempo, consejos y palabras de aliento para lograr este objetivo. Y en general a toda la Escuela de Computación e Informática por todo lo aprendido durante el periodo universitario; y por el apoyo que hasta ahora me siguen brindando.

Cinthya Viviana Amaya Pereyra

Agradecer a Dios por la vida, la cual me permite ahora estar disfrutando del logro de un objetivo más, a mi amada familia y en especial a mis padres Ana María Ecurra Alarcón y Amador Rodas Salazar quienes son mi inspiración y modelo por seguir. Agradezco a nuestro asesor Ing. Martín Leiva por su constante e inigualable apoyo.

Christian Cesar Rodas Ecurra

RESUMEN

La presente tesis, analiza la problemática central que enfrenta el país respecto al seguro agrícola catastrófico, el cual debido a la falta de integración de información entre la aseguradora que es POSITIVA SEGUROS Y REASEGUROS, MINAGRI, DRA'S y los peritos en este caso AGROEVALUACIONES PERÚ. Nuestro trabajo tiene como propósito, implementar un sistema informático el cual incorpore la información por dichas entidades permitiendo de manera oportuna, relevante y confiable mejorar la toma de decisiones a través de un sistema de gestión, análisis y cartografía para el seguro agrícola catastrófico.

Nuestro sistema funcionará en web y en dispositivos móviles Android, para lo cual hemos decido utilizar la metodología XP y los lenguajes Java y Python bajo el framework Django, los cuales nos permitirán realizar un sistema intuitivo, confiable y eficiente.

Este sistema incluye cartografía, generación aleatoria de los puntos de evaluación de los cultivos, manejo de estados de los avisos, generación de acta de evaluación, dictamen del aviso indemnizable, no indemnizable y desestimado, soporte a fotos de los cultivos evaluados, los peritos utilizarán una tablet para su inspección en campo y el resto de las entidades podrán realizar el seguimiento pertinente desde el sistema web.

Se logró optimizar un 84% el tiempo de atención e información directa con todos los usuarios, esto nos ayudó a reforzar la idea de que la implementación del sistema de gestión, análisis y cartografía para el seguro agrícola catastrófico fue realizada de manera correcta y oportuna para facilitar los trabajos en campo y la comunicación entre los usuarios.

INTRODUCCIÓN

En un mundo tan globalizado, donde la tecnología es el protagonista de la optimización de nuestro tiempo para las actividades de nuestra vida, ya sea personal o laboral, como son el uso de aplicativos para obtener algún servicio o producto, ubicación, citas médicas, ERP's, etc. Es importante saber que en el mundo de la agricultura también está implementando softwares que permiten a través de los satélites recopilar información la cual sea utilizada de manera eficiente y oportuna para la toma de decisiones.

En el Perú existe un marco legal del seguro agropecuario, el cual brinda apoyo del Ministerio de Agricultura y Riego (MINAGRI) a los productores de escasos recursos, el cual cubre riesgos climáticos catastróficos que afecten todo o casi todo lo sembrado.

Por lo que podemos decir que su objetivo principal es reducir los riesgos de origen climático, las vulnerabilidades y los efectos negativos del cambio climático en el sector agrario, a través de estrategias, lineamiento de políticas y acciones consensuadas con las regiones.

Ante la necesidad de un software que les ayude a cumplir sus objetivos, se realiza un sistema de gestión, análisis y cartografía para el seguro agrícola catastrófico; el cual les brindará optimización de tiempo y estar informados de todos los siniestros registrados por sus DRA's, registrando el tipo de evento, región afectada y la pérdida de producción, lo cual mantendrá un historial que permita tomar medidas preventivas en el sector agrario.

CAPÍTULO I: DISEÑO TEÓRICO

1. REALIDAD PROBLEMÁTICA

a. SITUACIÓN PROBLEMÁTICA INTERNACIONAL

La Implementación de los seguros Agrarios a nivel internacional se rigen bajo las condiciones del estado, mediante decretos, resoluciones o documentos que indiquen los criterios gubernamentales para la asignación de las subvenciones al precio del seguro de las diferentes producciones a las que el sistema da protección.

En todos los países, el seguro agrario tiene las principales fallas en común:

1. Fallas estructurales en la estrategia y en los procesos de información sobre el producto a los pequeños y medianos agricultores: el beneficiario no tiene la información necesaria y el conocimiento adecuado para ser parte de este sistema; excesiva documentación asociada al producto.
2. Inexistencia de un trabajo articulado entre aseguradora, institución financiera del estado y los peritos que realizan la evaluación.
3. Falta de control documental del proceso, ya que, en la mayoría de los países, aún se trabaja con documentos en físico.
4. Pésimo proceso de liquidación de siniestros, tardan demasiado en generar un dictamen.
5. Al no contar con la información en las tomas de decisiones para asuntos importantes que requieren atención inmediata.

6. Seguros agrarios se toma como el instrumento más adecuado para hacer frente a "las consecuencias derivadas de acontecimientos catastróficos". En algunos países se definió la posición entre "seguros o ayudas directas" ante daños de naturaleza catastrófica, que fue el objeto de controversia desde principio de siglo. Dejando la ayuda directa solo para casos de catástrofes muy puntuales. En países como España la cobertura más completa la tiene el llamado seguro integral, el cual sólo es otorgado para cultivos de invierno y oleaginosas, los vinos de Rioja y Lanzarote, las cebollas de Lanzarote y comprende heladas, granizo, inundaciones, sequía enfermedades y plagas. El alcance de las coberturas para otros cultivos es limitado y usualmente comprende heladas, tormentas, granizo e inundaciones. Para cultivos especiales como frutillas, tabaco, vid la cobertura también incluye daños por lluvias. En Panamá el Seguro Agrícola cubre los cultivos de arroz, café, caña de azúcar, cucurbitáceas (melón, sandía, zapallo), frijol, guandú, maíz, melón, ñame, otoi, palma aceitera, plátano, pimentón y tomate de mesa, porotos, sandía, yuca, zapallo, tomate industrial. Se basa en los costos directos de producción ante los riesgos que puedan afectarlos, ya que es una actividad que depende de los factores de la naturaleza y que el productor no puede predecir con facilidad. El problema que viene presentando es el reclamo de los productores ante la falta de pólizas adecuadas para su sector, pero al ser un seguro que requiere de un gran volumen de producción para ser rentable, las aseguradoras ven poco atractivo en el negocio. El seguro agrícola suele ser una de las muchas herramientas que los agricultores pueden utilizar como parte de su estrategia integral de gestión de los riesgos de la producción agrícola, la cual se utiliza principalmente para protegerse contra el riesgo de una pérdida de producción. Panamá y Nicaragua son los primeros países centroamericanos en contar con un sistema mixto de seguros agropecuarios.

b. SITUACIÓN PROBLEMÁTICA NACIONAL- LOCAL

El seguro agrícola catastrófico es un tipo de seguro que ya tiene cinco años trabajándose en el Perú, el cual en una misma póliza se protege grupos variados de cultivos (básicos, hortalizas, frutales y forrajes), amparando áreas extensas de cultivos pequeños o medianos, estableciendo un valor asegurado por hectárea igual para todos los cultivos protegidos y un rendimiento medio regional que fungirá como disparador conformados por productores para determinar la ocurrencia de catástrofe en una comunidad agrícola determinada (entendiéndose como catástrofe al daño que produce un fenómeno natural en la región lo cual da como consecuencia la pérdida parcial o casi total del cultivo) este disparador es el que nos indicará si el cultivo afectado es indemnizable o no, las aseguradoras que respaldan el seguro catastrófico con la POSITIVA SEGUROS Y REASEGUROS y MAPFRE PERÚ.

En la campaña 2016 -2017 se tienen aseguradas las regiones de Ayacucho, Puno, Pasco y Cajamarca por parte de la aseguradora la positiva, otra aseguradora tiene las regiones de Apurímac, Huancavelica, cusco y Huánuco, el seguro catastrófico cubre los siguientes eventos sequía, inundación, huaico o deslizamiento, temperaturas extremas, granizos y vientos fuertes.

los problemas que viene presentando el seguro agrícola catastrófico actualmente son los siguientes:

1. Los procesos que manejan son repetitivos y ocasionan doble trabajo.
2. La información no puede ser visualizada de manera lineal por todos los usuarios involucrados. lo cual les permita saber en qué estado se encuentra el aviso.
3. La información llega con desfase de 15 días a más a los usuarios involucrados para el pago de dichos seguros, lo cual impide tomar decisiones correctas.

4. Las evaluaciones en campo muchas veces son cuestionadas por su veracidad en la información. no tienen fotos del cultivo asegurado que corrobore esa información.
5. Existen molestias por parte del MINGARI respecto a la ubicación de los 11 puntos de muestra que se toman como referencia para evaluar el sector estadístico.
6. Tanto la aseguradora como el MINAGRI tienen la presión de beneficiarios que quieren saber si su cultivo es indemnizable o no; lo cual por el tema tiempos de envío de la información no tienen una respuesta.
7. Los evaluadores a campo tienen que llevar todos sus materiales como GPS, palas, balanzas y actas en físico para poder realizar dicha evaluación. se quiere evitar el uso del papel y que todo sea virtualizado.

El problema central que presenta el seguro agrario catastrófico se debe a la falta de integración de información entre la aseguradora que es POSITIVA SEGUROS Y REASEGUROS, MINAGRI, DRA'S y los peritos en este caso AGROEVALUACIONES PERÚ, por lo que se propone implementar un sistema informático el cual incorpore la información por dichas entidades permitiendo de manera oportuna, relevante y confiable mejorar la toma de decisiones.

2. FORMULACIÓN DEL PROBLEMA

¿Permitirá el desarrollo del Sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico minimizar los tiempos en la toma de decisiones por parte de La Positiva Seguros y MINAGRI?

3. OBJETO DE ESTUDIO Y CAMPOS DE ACCIÓN

- a) **Área** : Computación e Informática
- b) **Sub área** : Aplicada

c) **Línea** : Desarrollo de Software

d) **Programa** : Informática aplicada a la calidad Total

4. DELIMITACIÓN DE LA INVESTIGACIÓN

El sistema se desarrollará para Agroevaluaciones Perú E.I.R.L, con todos los requerimientos solicitados por esa empresa, se contará con la participación de parte de La Positiva Seguros y Reaseguros, MINAGRI, FOGOSA y las DRA'S que son los usuarios que nos brindarán sus propios requerimientos para hacer que este sistema sea productivo y se ajuste a las necesidades del seguro Agrario Catastrófico en el Perú y les permita tomar las mejores decisiones ante un siniestro catastrófico. El sistema permitirá que las DRA'S ingresen los avisos catastróficos al sistema y les permita escoger los puntos referenciales de evaluación de manera aleatoria, garantizando que las evaluaciones se realizarán de acuerdo con la metodología que ellos manejan para este cálculo, estos puntos estarán reflejados en el mapa que genera el sistema cuando se registra el siniestro catastrófico. La información ingresada por las DRA'S podrán ser observadas, analizadas y verificadas por todos los usuarios que intervienen en este sistema, siguiendo paso a paso los cambios de estado por los que pasa cada aviso. Se podrá visualizar las fotos por cada punto de evaluación, las actas firmadas por el agente agrario.

5. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN

a. ASPECTO TECNOLÓGICO

El estudio del proyecto servirá para aprovechar los recursos tecnológicos relacionados con las nuevas tendencias en tecnologías para optimizar los procesos del Seguro Agrícola Catastrófico. El desarrollo del Sistema Web que integrará a las entidades como La Positiva Seguros, MINAGRI, DRA'S y Agroevaluaciones Perú EIRL, va a contribuir a la eficiencia,

eficacia y productividad sean mayores en el Seguro Agrícola, haciendo uso de las propuestas adecuadas y las herramientas tecnológicas necesarias se agilizará procesos y consultas por parte de las entidades involucradas en el proceso.

b. ASPECTO ECONÓMICO

El estudio del proyecto busca reducir costos directos e indirectos tales como:

i. COSTOS DIRECTOS

- Con el desarrollo del Sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico, La Positiva Seguros evitará multas impuesta por el MINAGRI; al no cumplir con los tiempos en la generación de los dictámenes (Indemnizables y no Indemnizables), según las evaluaciones brindadas por los Peritos Agroevaluaciones Perú.
- Obtener información online oportuna y confiable sobre los procesos.

ii. COSTOS INDIRECTOS

- Los Pagos que se realizarán a los pequeños agricultores sea de manera oportuna y sobre todo que cuenten con el sustento de evaluación como son las actas firmadas por las DRA'S, fotos de las condiciones en las que se encontró el cultivo.

iii. ASPECTO SOCIAL

- El estudio del proyecto ayudará para uniformizar la información del Seguro Agrícola Catastrófico en el Perú, permitiendo de manera oportuna, relevante y confiable mejorar la toma de decisiones. Así también permitirá estar a la vanguardia en el uso de tecnologías de la información y generar reportes para obtener un dictamen justo para los pequeños agricultores.

6. LIMITACIONES DE LA INVESTIGACIÓN

Al realizar este proyecto de investigación se encuentra que las limitantes de que tenemos que considerar son las siguientes:

- El sistema web y móvil necesita una conexión a Internet, los evaluadores antes de ir a campo deberán descargar los avisos a sus tablets accediendo a internet, la información que registren en campo no necesita dicha conexión para guardar los datos, pero para enviar al servidor tienen que estar conectadas a internet obligatoriamente.
- Los mapas cartográficos deben ser entregados por el Ministerio de Agricultura y Riego en formato .mdb, el cual nos permite la correcta sincronización para generar los bastones y puntos aleatorios de evaluación.

Éstas son las limitaciones que encontramos para que el Sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico funcione correctamente.

7. OBJETIVOS DE LA INVESTIGACIÓN

a. OBJETIVO GENERAL

- Desarrollar un Sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico para minimizar los tiempos en la toma de decisiones para La Positiva y el MINAGRI.

b. OBJETIVO ESPECÍFICO

1. Planificar iteraciones e historias de usuario, según la necesidad de la empresa para la ejecución del Sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico.

2. Diseñar todas las iteraciones elegidas por el usuario de manera sencilla e intuitiva, logrando mejorar la experiencia de los usuarios MINAGRI, LA POSITIVA SEGUROS y AGROEVALUACIONES PERÚ EIRL.
3. Codificar las historias de usuario de forma sencilla con la participación fluida del cliente y así obtener resultados apropiados para el MINAGRI, LA POSITIVA SEGUROS y AGROEVALUACIONES PERÚ EIRL.
4. Garantizar la confiabilidad del sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico y el cumplimiento de los requisitos solicitados por medio de pruebas unitarias y de integración.

8. ASPECTO DE LA ORGANIZACIÓN

Tenemos que considerar, que las entidades que intervienen en los procesos del sistema son: MINISTERIO DE AGRICULTURA Y RIEGO, DIRECCIÓN REGIONAL DE AGRICULTURA, POSITIVA SEGUROS Y REASEGUROS Y AGROEVALUACIONES PERÚ SRL, por lo cual tenemos que abarcar la información de las cuatro entidades para los siguientes ítems:

a. NATURALEZA

i. MINISTERIO DE AGRICULTURA Y RIEGO

(MINAGRI, MINAGRI, 2019) El Ministerio de Agricultura y Riego es un organismo del Poder Ejecutivo, ente rector en materia agraria, con personería jurídica de derecho público, y constituye un Pliego Presupuestal. El MINAGRI, ejerce su competencia a nivel nacional, en las siguientes materias:

- Tierras de uso agrícola y de pastoreo, tierras forestales y tierras erizas con aptitud forestal.
- Recursos forestales y su aprovechamiento.

- Flora y fauna.
- Recursos hídricos.
- Infraestructura agraria.
- Riego y utilización de agua para uso agrario.
- Cultivos y crianzas.
- Sanidad, investigación, extensión, transferencia de tecnología y otros servicios vinculados a la actividad agraria.

ii. DIRECCIÓN REGIONAL DE AGRICULTURA

(MINAGRI, MINAGRI, 2019) La Dirección Regional Agraria es la instancia principal de coordinación a nivel Regional, de las actividades del Ministerio de Agricultura, constituye la segunda instancia en materia administrativa. La Dirección Regional de Agricultura, es la encargada de promover, supervisar, evaluar, dirigir y controlar las diferentes actividades productivas agrarias en el ámbito de la Región. Ejerce autoridad jerárquica y administrativa sobre las unidades orgánicas de la Dirección Regional de Agricultura. Está a cargo de un funcionario con categoría de Director Regional, quien es el representante institucional del Sector Agrario a nivel regional.

iii. LA POSITIVA SEGUROS Y REASEGUROS

(Positiva, 2019) La Positiva Seguros es un Grupo Asegurador con más de 77 años de experiencia en el mercado asegurador peruano y cuenta con más de 3'500,000 de asegurados en todo el Perú. La Positiva Seguros está conformada por:

- La Positiva Seguros Generales, que ofrece seguros de bienes patrimoniales, accidentes y de salud.
- La Positiva Vida, que ofrece seguros de vida, accidentes de trabajo, seguros previsionales y renta de jubilación.

- La Positiva EPS, que ofrece Planes de Salud y cobertura médica al Seguro Complementario de Trabajo de Riesgo.
- Seguro Agrario Catastrófico, está encaminado a proteger los cultivos de los productores agropecuarios, principalmente de escasos recursos, ante la presencia de riesgos de peculiaridades catastróficas, entendiendo como catástrofe al daño producido de impacto regional y que tiene como consecuencia la pérdida total o casi total de la especie agropecuaria producida en una zona geográfica determinada. (Positiva, 2019)

iv. AGROEVALUACIONES PERÚ SRL

AGROEVALUACIONES SRL cuenta con una merecida reputación por un servicio al cliente de gran calidad, es una empresa dedicada al rubro de evaluaciones agrarias, está orientada al seguro agrícola catastrófico y el seguro comercial, la naturaleza de esta empresa es evaluar cultivos que presenten un siniestro, en el cual tienen que identificar la fenología y el tipo evento por el cual está siendo afectado el cultivo ,cuánto es el daño que está sufriendo la parcela en general y su producción actual. (Agroevaluaciones, 2019)

b. JURISDICCIÓN

i. MINISTERIO DE AGRICULTURA Y RIEGO

La política nacional de modernización de la gestión pública¹ , establece que las políticas públicas nacionales y el planeamiento se constituyen como uno de los cinco (05) pilares centrales para concretar una gestión pública por resultados, que impacte positivamente en el bienestar del ciudadano y el desarrollo del país, determinando que, “las Políticas Públicas Nacionales se enmarcan en Políticas de Estado, y responden a un Programa de Gobierno y a

¹ Aprobada mediante Decreto Supremo N° 004-2013-PCM.

la identificación de los problemas de la agenda pública, que deben priorizarse tomando en cuenta las necesidades o demandas ciudadanas”. (MINAGRI, MINAGRI, 2019)

En virtud de ello, el Ministerio de Agricultura y Riego, dada sus facultades y competencias, aprobadas mediante la Ley N° 30048², ha diseñado la Política Nacional Agraria, como resultado de la difusión y debate entre múltiples actores vinculados al sector agrario, el que consideró al sector público, expertos, colegio de profesionales, organizaciones de productores, empresas, entre otros.

Este proceso de consulta pública se llevó a cabo entre los meses de abril y julio de 2015, mediante publicación efectuada en el portal institucional y a través de seis (06) talleres macrorregionales³ realizados en Ica, Cusco, San Martín, Arequipa, Junín y Lambayeque, y dieciocho (18) talleres regionales⁴. En conjunto se registró la participación de representantes de los veinticinco (25) Gobiernos Regionales, de las Gerencias Regionales de Agricultura y Direcciones Regionales Agrarias, Agencias Agrarias, Juntas de Usuarios de los Distritos de Riego, alcaldes, investigadores, organizaciones de productores, entre otros.

Como resultado, se ha construido la Política Nacional Agraria, la cual consta de doce (12) Ejes de Política que generan un marco orientador de mediano y largo plazo que favorecerán el desarrollo sostenible de la agricultura, con prioridad en la agricultura familiar, y permitirán activar el desarrollo y la inclusión social en beneficio de la población rural, contribuyendo a la seguridad alimentaria y nutricional en el Perú.

² Ley que modifica el Decreto Legislativo N° 997, Decreto Legislativo que aprueba la Ley de Organización y Funciones del Ministerio de Agricultura.

³ Ica – 9 y 10 de abril (con la asistencia de los representantes de Ica, Lima provincia, Ayacucho, Huancavelica y Ancash); Cusco – 16 y 17 de abril (con la asistencia de los representantes de Cusco, Madre de Dios, Apurímac, Puno); San Martín – 21 y 22 de abril.

⁴ Lima, Puno, Ayacucho, Cajamarca, Ucayali, Loreto, Huancavelica, Ancash, Pasco, Huánuco, Amazonas, Piura, Moquegua, Tumbes, Tacna, Madre de Dios, Apurímac y La Libertad. A nivel de los talleres macrorregionales y regionales participaron más de 1,600 personas.

ii. DIRECCIÓN REGIONAL AGRARIA

Las direcciones regionales agrarias son organismos oficiales en la Región para cubrir aspectos de sanidad agrícola y pecuaria, su misión es promover y desarrollar la actividad agraria en un entorno interinstitucional, sobre la base de la equidad, gestión sostenible y competitiva de los recursos naturales, en el marco de la modernización y descentralización de la Región; para contribuir a mejorar la calidad de vida del poblador y garantizar la seguridad alimentaria

iii. LA POSITIVA SEGUROS Y REASEGUROS

La Compañía tiene por objeto dedicarse a la contratación y administración de toda clase de seguros dentro de los alcances de la Ley No.26702, Ley General del Sistema Financiero y del Sistema de Seguros y Orgánica de la SBS (en adelante la Ley General). En adición, percibe ingresos por el alquiler de inmuebles de su propiedad e intereses y dividendos por inversiones efectuadas libremente y/o de acuerdo con disposiciones emitidas por la SBS. (Positiva, 2019)

iv. AGROEVALUACIONES PERÚ SRL

AGROEVALUACIONES PERÚ SRL, empresa peruana que brinda servicios de evaluación a los cultivos asegurados por la empresa LA POSITIVA SEGUROS Y REASEGUROS, participando como peritos, brindando la información pertinente para la indemnización por los daños que pueda presentar cada cultivo asegurado. (Agroevaluaciones, 2019)

c. DOMICILIO LEGAL

i. MINISTERIO DE AGRICULTURA Y RIEGO

Sus oficinas de encuentran ubicadas en las Av. La Universidad N°200, Av. Alameda del Corregidor N°155 y Jr. Yauyos N°258 en el distrito de La Molina – Lima- Perú.

ii. DIRECCIÓN REGIONAL AGRARIA

- Dirección Regional Agraria Ayacucho
 - Dirección: Av. Independencia 604
- Dirección Regional Agraria Cajamarca
 - Dirección: Carretera Cajamarca baños del Inca Km 3.5 Bellavista.
- Dirección Regional Agraria Puno
 - Dirección: Jr. Moquegua N° 264
- Dirección Regional Agraria Pasco
 - Dirección: Jr. Calicanto 145

iii. LA POSITIVA SEGUROS Y REASEGUROS

La Positiva Seguros cuenta con una amplia red de agencias a lo largo del territorio nacional, su sede principal se encuentra ubicada en:

- Esq. Javier Prado Este y Francisco Masías 370. San Isidro, Lima - Perú

iv. AGROEVALUACIONES PERÚ SRL

Tiene como dirección fiscal:

- Sector Dallorso Sur Lote. 08 (Espaldas Cementerio el Ángel) – Pimentel – Chiclayo.

d. DESCRIPCIÓN

i. MINISTERIO DE AGRICULTURA Y RIEGO

El Ministerio de Agricultura con ruc 20131372931, fue creado bajo la Ley N.º 9711, formulada el 31 de diciembre del año 1942, durante el gobierno del presidente, Dr. Manuel Prado Ugarteche y en un contexto de conflicto global, pues acontecía la Segunda Guerra Mundial.

En el marco de su creación el Ministerio de Agricultura contó con las Direcciones de Agricultura, Ganadería, Aguas e Irrigación, Colonización y Alimentación Nacional.

El presidente Prado Ugarteche dictaminó, a través de la mencionada ley, que se autorizase abrir los créditos necesarios para la vigencia del portafolio y que el Poder Ejecutivo velase por el funcionamiento de las diversas reparticiones del Ministerio y de sus Secciones, “indispensables para el estudio de los problemas de la costa, sierra y montaña, de acuerdo con las necesidades de las diversas regiones del país”, como lo indica la ley textualmente.

El 2 de enero de 1943 fue firmada La Ley N° 9711 y fue publicada en el diario oficial El Peruano el 26 de febrero de 1943. El flamante Ministerio de Agricultura entra en funciones el 1 de junio de 1943, siendo su primer ministro el Sr. Benjamín Roca García.

ii. DIRECCIÓN REGIONAL DE AGRICULTURA

Antes del proceso de descentralización, las DRA’S constituían para el MINAGRI los órganos regionales sobre el cual recaía directamente la responsabilidad de la ejecución de la política agraria en el ámbito regional.

Estas DRA’S a su vez estaban conformadas por las agencias agrarias en cada provincia, cubriendo así todo el ámbito Nacional. Es decir, el MINAGRI era identificado en el campo a través de las DRA’S. Con el proceso de descentralización, los gobiernos regionales no necesariamente los han fortalecido, dando lugar a que algunas dependencias del MINAGRI generaran oficinas zonales o regionales para la ejecución de políticas agrarias.

Sin embargo con las DRA’S se coordina algunos temas, como los de estadísticas, promoción, planificación, apoyándose en la formulación de sus planes estratégicos Regionales del Sector Agrario (PERSA). Las direcciones Regionales Agrarias, son las encargadas de atender los reclamos, solicitudes o siniestros que reporta cada agricultor de su región. (MINAGRI, Seguro Agrícola Catastrófico, 2019)

iii. LA POSITIVA SEGUROS Y REASEGUROS

La Positiva Seguros y Reaseguros (en adelante la Compañía) con ruc 20100210909, se constituyó en el Perú el 27 de setiembre de 1937. La Compañía es una sociedad anónima con domicilio legal en Calle San Francisco No.301, Arequipa en ese entonces; ejerce su actividad económica principalmente en la ciudad de Lima. La Positiva Seguros y Reaseguros S.A. y Subsidiarias (en adelante el Grupo), refleja al 31 de diciembre de 2015 la consolidación de los estados financieros de la Compañía, de su subsidiaria directa: La Positiva Vida Seguros y Reaseguros S.A. (en adelante Positiva Vida) y de sus subsidiarias indirectas Transacciones Financieras S.A., Alianza Seguros y Reaseguros S.A. y Alianza Vida Seguros y Reaseguros S.A. (estas dos últimas ubicadas en Bolivia). Al 31 de diciembre de 2014, incluyó la consolidación de los estados financieros de su subsidiaria Inversiones en Salud S.A. Con fecha 29 de agosto de 2014, la Superintendencia de Banca, Seguros y Administradoras Privadas de Fondos de Pensiones (en adelante la SBS) comunicó a la gerencia de la Compañía, mediante Oficio No.31558 - 2014 - SBS, una nueva estructura para el Grupo Consolidable Positiva. Como resultado de este análisis el Grupo Consolidable de La Positiva quedó conformado por dos subsidiarias directas, Positiva Vida e Inversiones en Salud S.A. y tres subsidiarias indirectas Transacciones Financieras S.A., Alianza Seguros y Reaseguros S.A. y Alianza Vida Seguros y Reaseguros S.A.

En el 2016 hacia adelante gestionaron La Positiva Seguros bajo una única visión corporativa, aprovechando la sinergia entre La Positiva Seguros y Reaseguros y La Positiva Vida. (Positiva, 2019)

iv. AGROEVALUACIONES PERÚ SRL

AGROEVALUACIONES PERÚ SRL, esta empresa fue fundada el 18/11/2014, registrada dentro de las sociedades mercantiles y comerciales como UNA SOCIEDAD CON RESPONSABILIDAD LIMITADA

e. ORGANIGRAMA

i. MINISTERIO DE AGRICULTURA Y RIEGO

- En nuestra imagen 1 mostramos como está estructuralizado el MINAGRI:

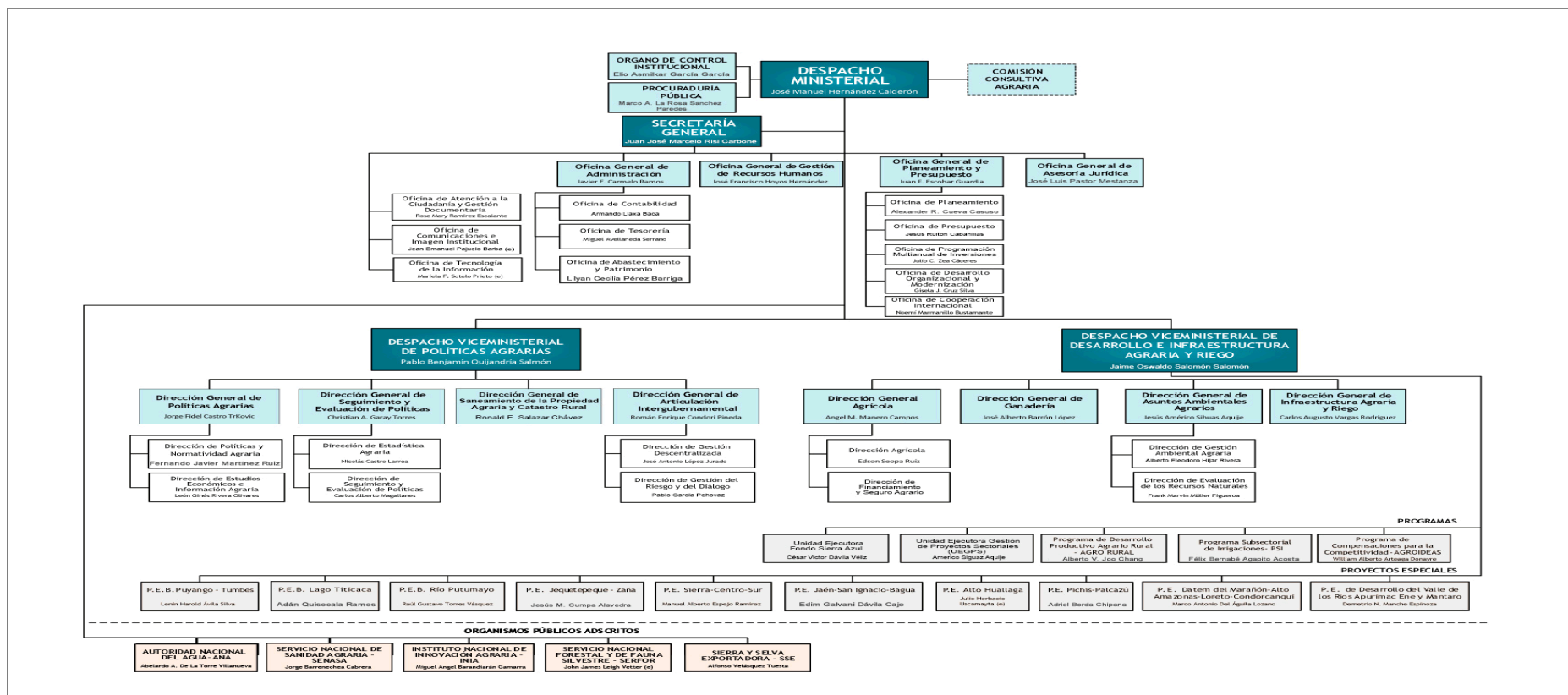


Imagen 1: Organigrama de Ministerio de Agricultura y Riego (MINAGRI)

Fuente: Información brindada por el cliente

ii. DIRECCIÓN REGIONAL AGRARIA

- En la imagen 2 mostramos como está estructuralizado las DRA'S:

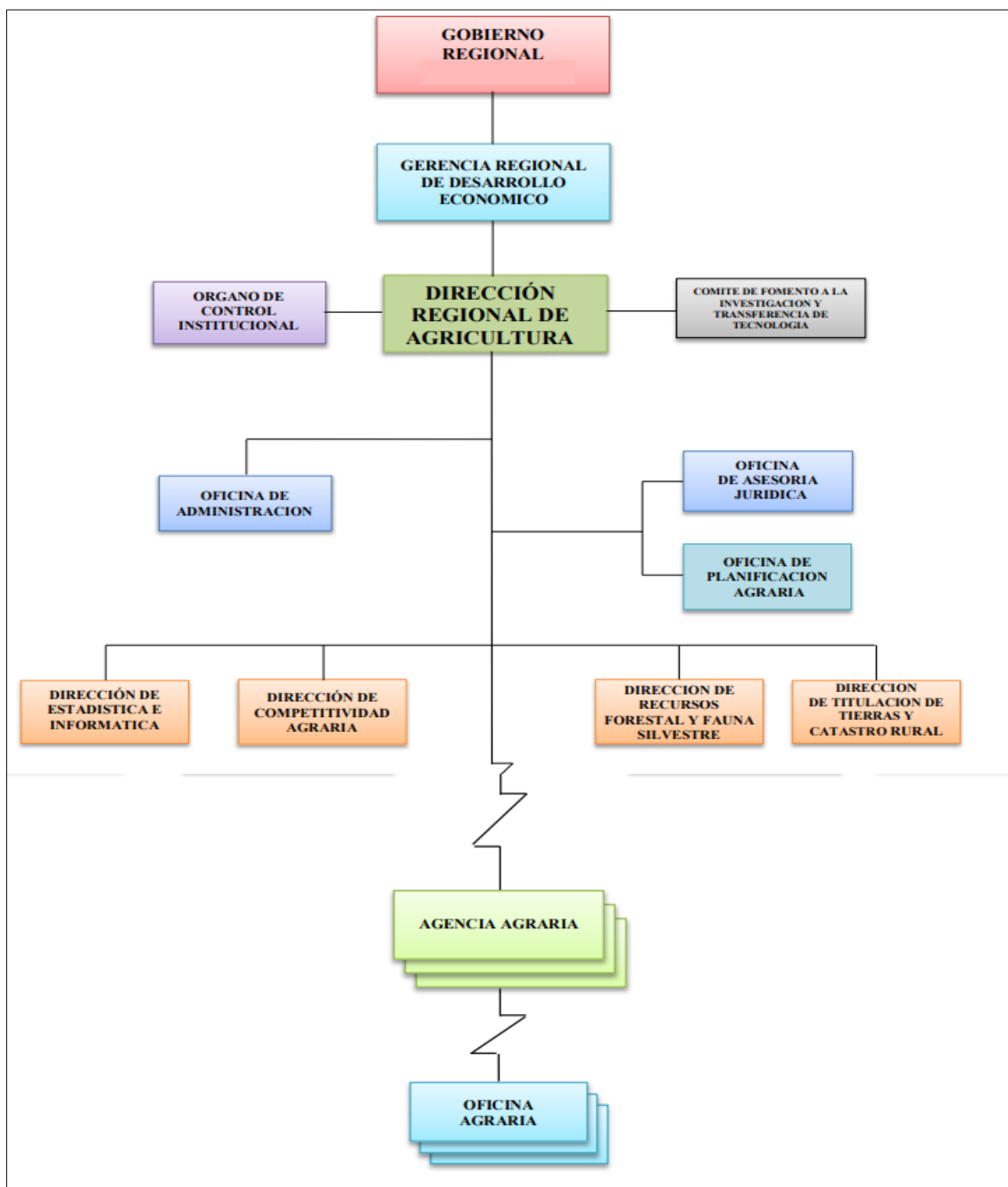


Imagen 2: Organigrama Dirección Regional Agraria

Fuente: Información brindada por el cliente

iii. LA POSITIVA SEGUROS Y REASEGUROS

En la imagen 3 mostramos como está estructuralizado La Positiva Seguros:

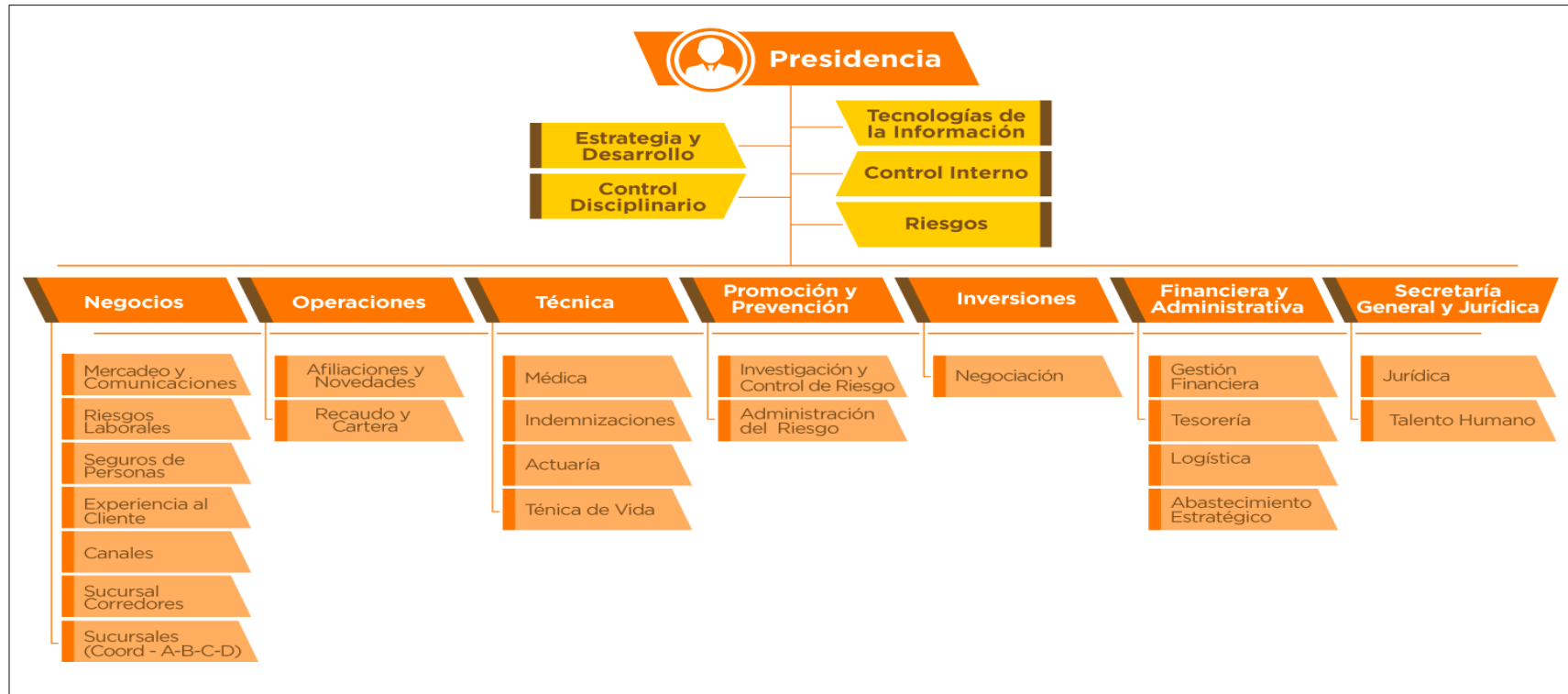


Imagen 3: Organigrama La Positiva Seguros

Fuente: Información brindada por el cliente

iv. AGROEVALUACIONES PERÚ SRL

En la imagen 4 mostramos como está estructuralizado Agroevaluaciones Perú SRL:

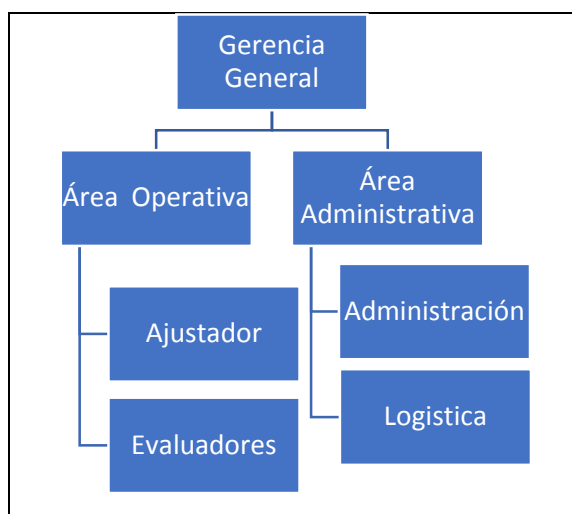


Imagen 4: Organigrama Agroevaluaciones Perú SRL

Fuente: Información brindada por el cliente

f. MISIÓN Y VISIÓN

i. MISIÓN

1. MINISTERIO DE AGRICULTURA Y RIEGO

Diseñar y ejecutar políticas para el desarrollo de negocios agrarios y de la agricultura familiar, a través de la provisión de bienes y servicios públicos de calidad.

2. DIRECCIÓN REGIONAL AGRARIA

Promover el desarrollo integral y sostenible de las regiones, con autonomía e igualdad de oportunidades, en el marco de la gestión pública moderna, transparente, competitiva y concertada, orientado a mejorar la calidad de vida de la población.

3. LA POSITIVA SEGUROS Y REASEGUROS

Ser la mejor opción del mercado asegurador.

4. AGROEVALUACIONES PERÚ SRL

Ser una empresa líder en las evaluaciones agrarias, actuando con justicia e imparcialidad.

ii. VISIÓN

1. MINISTERIO DE AGRICULTURA Y RIEGO

Al 2021, Perú tiene un agro próspero, competitivo e insertado al mercado nacional e internacional, a través de la productividad y calidad de sus productos agroalimentarios.

2. DIRECCIÓN REGIONAL AGRARIA

Ser un organismo líder en el desarrollo agrario de la región, asentándose estratégicamente en los mercados Nacionales e Internacionales con productos ecológicos y biodiversos obtenidos de una gestión participativa, pública, privada, moderna, competitiva buscando permanentemente el fortalecimiento y coordinación de instituciones y organizaciones rurales.

3. LA POSITIVA SEGUROS Y REASEGUROS

Somos una empresa experta en gestión de riesgos enfocada en lograr una alta penetración de mercado a través de nuestros productos innovadores, utilizando múltiples canales de acceso, con excelencia y rapidez de respuesta en nuestro servicio.

4. AGROEVALUACIONES PERÚ SRL

Organizar la información de los siniestros agrícolas y hacerla universalmente accesible y útil para una intervención oportuna de las aseguradoras y beneficiar a los pequeños y grandes agricultores.

9. ANTECEDENTES DE ESTUDIO

a. A NIVEL INTERNACIONAL

Rol de los Sistemas de Información Geográfica (SIG) dentro de un plan de ordenamiento territorial y desarrollo a nivel cantonal con un enfoque geo-administrativo. (Universidad San Francisco De Quito - Ligia Paulina Pérez Yánez - 2016)

se propone una metodología la cual brinda un plan de ordenamiento territorial y desarrollo a nivel cantonal con un enfoque geo administrativo, utilizando herramientas SIG, en la cual se integra variables en los componentes biofísicos, económicos y sociales, las cuales permiten tener un mayor entendimiento del comportamiento territorial. se deduce que el análisis del medio social con SIG puede establecer dónde localizar o reubicar los asentamientos (población e infraestructura de vías y servicios), lo que constituye ventajoso en la planificación, logrando que la distribución de los servicios sea adecuada y llegue a toda la población. En el medio económico, establece los tipos sistemas de producción representativos de la zona de estudio, identifica las actividades productivas y optimiza recursos, a más de esto identifica la dinámica productiva y se articula a los mercados e infraestructura de la producción. y en el medio biofísico permite conocer cuáles son las zonas de recursos naturales, relieve, hidrografía, precipitación, suelo, uso y cobertura del suelo, zonas de protección forestal, cobertura natural.

mediante este proyecto se propone que dentro de un SIG se debe conservar la base de datos, estructurada, organizada, detallada, explicativa, ajustada a lineamientos geográficos como catálogo de objetos, para retroalimentar y generar nuevas líneas de investigación en cuanto a la gestión del territorio. (Ligia, 2016)

b. A NIVEL NACIONAL

Desarrollo de un Módulo SIG para el manejo de Imágenes Multiespectrales orientado a la agricultura de precisión. (Pontificia Universidad Católica del Perú - Mario Balcázar Guerrero – 2011).

Se busca desarrollar una interfaz que haga uso de las herramientas propias de los sistemas de información geográfica, en particular del SIG GRASS, para la integración y el posterior análisis de imágenes multiespectrales de cultivos adquiridas mediante el uso de un sistema de censado remoto o la teledetección que es la técnica de adquisición y posterior tratamiento de datos de la superficie terrestre desde sensores instalados en plataformas espaciales (aeronaves o satélites), en virtud de la interacción electromagnética existente entre la tierra y el sensor, siendo la fuente de radiación proveniente del sol (teledetección pasiva) o del propio sensor (teledetección activa), que fue desarrollado en la Pontificia Universidad Católica del Perú, con las cuales se permita evaluar la situación actual de los mismos, en base al índice vegetativo conocido como factor NDVI, que es una fórmula usada para medir cuantitativamente el crecimiento de las plantas y la producción de biomasa a partir de datos satelitales multiespectrales. Se concluye que el software SIG GRASS permite procesar las imágenes adquiridas tanto del censado remoto como las imágenes adquiridas desde un satélite, y el método por el cual se crea una imagen GEOTIFF que integra la información de las imágenes multiespectrales adquiridas con sus respectivos datos GPS, permite ingresar ambos datos al sistema de información para su posterior manipulación. En lo que respecta al factor NDVI, nos permite diferenciar entre la vegetación sana y otros elementos presentes en las imágenes adquiridas, tales como suelo árido o incluso la presencia de zonas cubiertas de agua. A través de los comandos y los módulos de GRASS se obtiene la información de interés; es decir, los valores NDVI y las coordenadas geográficas asociadas a las imágenes analizadas en dicho sistema de manera visual. (Mario, 2011)

c. A NIVEL LOCAL

Elaboración de cartografía de riesgo de inundaciones y propuesta de mejora de la conservación en la cuenca y Reserva de Mar Chiquita, Provincia de Buenos Aires, Argentina. (Universidad Internacional De Andalucía- Lic. Juan Pablo Celemín -2009)

La cuenca superficial de Mar Chiquita, con alrededor de un millón de hectáreas (en adelante has.), constituye una unidad ecológica donde se combinan los diferentes usos de la tierra presentes en el resto de la región pampeana. La zona núcleo de la reserva comprende unas 5000 has. cubiertas por la laguna y unas 19000 has. de campos aledaños. La zona de estudio se caracteriza por presentar, en forma periódica y frecuente, inundaciones prolongadas, situación que se ha agravado desde hace treinta años a partir del inicio de un período húmedo con consecuencias negativas para toda el área. Con la utilización de imágenes satelitales y SIG se logró crear, analizar y combinar diferentes mapas que permitieron obtener el producto cartográfico final. El mismo da origen a cuatro categorías de riesgo (alto, medio, bajo y nulo) que permiten distinguir la importancia de la unidad geomorfológica Franja Eólica Peri serrana (por poseer los mejores suelos y los centros urbanos) y de las vías de comunicación en general. Por lo tanto, estas zonas de la Cuenca Mar Chiquita son a las que les correspondieron los mayores índices de riesgo. Este mapa resalta la importancia de las nuevas tecnologías aplicadas como herramientas para una mejor interpretación y análisis del territorio a diferentes escalas de trabajo. En consecuencia, servirá como base para elaborar una propuesta de ordenamiento territorial de toda a la zona. El método empleado para desarrollar el mapa de riesgo de inundaciones se denomina Multi Criterio Evaluación (MECE), su capacidad se nota muy especialmente en aquellos mapas donde existan valores continuos (distancias o grados) de los cuales se pueden extraer distancias. (Juan, 2009)

CAPÍTULO II: MÉTODOS Y MATERIALES

1. BASE TEÓRICA

a. SISTEMAS DE INFORMACIÓN GEOGRÁFICA SIG

El término de sistemas de información geográfica (SIG⁵), es un estándar que permite dibujar cartografía⁶, procesamiento de datos geográficos como es el GEOPOSICIONAMIENTO, que permite realizar cálculos de áreas, análisis de información y manejar diferentes sistemas de mediciones como latitud, longitud, UTM⁷. En resumen, podemos indicar que el SIG es un conjunto de herramientas diseñadas para obtener, almacenar, recuperar y desplegar datos espaciales del mundo real. “Un SIG es un elemento que permite analizar, presentar e interpretar hechos relativos a la superficie terrestre”. El mismo autor indica, que “Esta es una definición muy amplia, y habitualmente se emplea otra más concreta en palabras habituales, un SIG es un conjunto de software y hardware diseñado específicamente para la adquisición, mantenimiento y uso de datos cartográficos”. (Tomlin, 1990)

2. ENFOQUES ÁGILES

a. MANIFIESTO ÁGIL

La historia nos indica que, el 17 de febrero de 2001, se reunieron diecisiete personas, convocados por Kent Beck, quien había publicado un par de años antes Extreme PROGRAMAN EXPLAINED, en el LODGE AT SNOWBIRD en las montañas Wasatch

⁵ Siglas en ingles Geographic, o Geographical, Information Systems (GIS).

⁶ Disciplina que trata sobre la concepción, producción, difusión y estudio de los mapas. (Asociación Cartográfica Internacional,1995).

⁷ UTM: Universal Transverse Mercator, se utiliza para referenciar cualquier punto de las superficies terrestres.

de Utah, para tratar sobre técnicas y procesos para desarrollar software. Lo que surgió fue el Manifiesto Agile 'Desarrollo de Software'.

Representantes de EXTREME PROGRAMMING, SCRUM, DSDM, Desarrollo de Software Adaptativo, CRYSTAL, FEATURE-DRIVEN DEVELOPMENT, PRAGMATIC PROGRAMMING, y otros simpatizantes que presentaban la necesidad y búsqueda de una alternativa para solucionar inconvenientes en la documentación de los procesos pesados que implicaba el desarrollo de software en ese entonces.

Los integrantes de la reunión resumieron los principios sobre los que se basan los métodos alternativos en cuatro postulados, lo que ha quedado denominado Manifiesto Ágil que indica lo siguiente:

“Estamos descubriendo las mejores formas de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.” (Cunningham, 2001)

Las personas que se reunieron son:

KENT BECK, MIKE BEEDLE, ARIE VAN BENNEKUM, ALISTAIR COCKBURN, WARD CUNNINGHAM, MARTIN FOWLER, JAMES GRENNING, JIM HIGHSMITH, ANDREW HUNT, RON JEFFRIES, JON KERN, BRIAN MARICK, ROBERT C. MARTIN, STEVE MELLOR, KEN SCHWABER, JEFF SUTHERLAND, DAVE THOMAS.

b. PRINCIPIOS DEL MANIFIESTO ÁGIL:

El manifiesto ágil, establece estos 12 principios:

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que se hace es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto organizan.
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

3. METODOLOGÍA POR USAR

a. ANÁLISIS DE LA METODOLOGÍA POR USAR

Antes de indicar que metodología usaremos para este proyecto, nos gustaría mostrar esta tabla 1 donde analizaremos mediante un comparativo para identificar las bondades de cada metodología:

CRITERIO	METODOLOGÍAS				
	eXtreme Programming (XP)	Test Driven Development (TDD)	Acceptance Test Driven Development (ATDD)	Behavior Driven Development (BDD)	Crystal Clear
Tipo de Iteraciones	Iteraciones de plazo variable: Los planes no son predicciones del futuro, sino simplemente la mejor estimación de cómo saldrán las cosas. Los planes son útiles, pero necesitan ser cambiados cuando las circunstancias lo requieren.	Guiado por pruebas: La generación de pruebas para cada funcionalidad hace que el programador confíe en el código escrito. Esto permite hacer modificaciones profundas del código (posiblemente en una etapa de mantenimiento del programa) pues sabemos que si luego logramos hacer pasar todas las pruebas tendremos un código que funcione correctamente.	Desarrollo Basado en Pruebas: Las pruebas de aceptación se crean cuando se analizan los requisitos y antes de la codificación. Pueden ser desarrollados en colaboración por el solicitante de requisitos (propietario del producto, analista comercial, representante del cliente, etc.), desarrollador y evaluador. Las pruebas se especifican en términos de dominio comercial.	Historias de Usuario: Identifican a un grupo de intereses, un efecto y un valor comerciales. También describe varios escenarios, cada uno con una condición previa, desencadenante y resultado esperado.	Iteraciones cortas con feedback (retroalimentación): Frecuente por parte de los usuarios/clientes, minimizando de esta forma la necesidad de productos intermedios. Son los requerimientos que se establecieron al inicio del proyecto. Consiste en entregar software a los clientes con frecuencia, no solamente en compilar el código. La frecuencia dependerá del proyecto, pero puede ser diaria, semanal o mensual.
Roles - Facilitador	Coach: El objetivo de un entrenador es que el equipo comprenda las directrices de XP. No se trata de que sean solamente lecciones teóricas, sino que se trata de dar ejemplo y propone ideas para mejorar. Big Boss: Es el gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado. El cliente puede asumir este papel.	N/A	Las pruebas son creadas por la tríada de cliente, probador y desarrollador . El probador y el desarrollador pueden crear más ejemplos y hacer que el cliente los revise.	Experto de Dominio : Es el Analista del Negocio, trabaja como un enlace entre los distintos interesados de un proyecto para obtener, analizar, comunicar y validar los requerimientos de cambios a los procesos de negocio, políticas y sistemas de información. El rol del analista de negocio es fundamental dentro de nuestras organizaciones, no solo porque gestiona los requerimientos, sino porque cierra la brecha que existe entre los interesados en el proyecto (stakeholders) y el equipo de desarrollo.	Patrocinador: Produce la Declaración de Misión con Prioridades de Compromiso (Tradeoff). Consigue los recursos y define la totalidad del proyecto. Coordinador: Con la ayuda del equipo, produce el Mapa de Proyecto, el Plan de Entrega, el Estado del Proyecto, la Lista de Riesgos, el Plan y Estado de Iteración y la Agenda de Visualización.
Roles - Administrador Requerimientos	Cliente: Es el responsable de conducir el proyecto. Define el proyecto y sus objetivos. Cuanto más preciso es su trabajo y cuanto mayor sea su involucración, mayores serán las oportunidades de éxito.	N/A	Cliente: Realizan pruebas que demuestran la intención comercial del sistema.	Cliente: Persona de negocios que especifica los comportamientos que desean ver en el sistema.	Usuario Experto: Junto con el Experto en Negocios produce la Lista de Actores Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe familiarizarse con el uso del sistema, sugerir atajos de teclado, modos de operación, información a visualizar simultáneamente, navegación. Experto en Negocios: Junto con el Usuario Experto produce la Lista de Actores Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe conocer las reglas y políticas del negocio.

Roles – Equipo Proyecto	<p>Programador: Una vez que se han comprendido las historias de usuario, el XP adjudica a los programadores la responsabilidad de tomar decisiones técnicas.</p> <p>Test: El encargado de pruebas ayuda al cliente a definir y escribir las pruebas de aceptación de las historias de usuario.</p>	N/A	<p>Desarrollador: Los desarrolladores implementan el sistema utilizando las pruebas de aceptación. Los desarrolladores conectan la prueba al sistema escribiendo fragmentos cortos de código.</p> <p>Probador: Genera la automatización de las pruebas de aceptación les permite ejecutarse como pruebas de regresión para garantizar que las nuevas características no interfieran con las características desarrolladas previamente.</p>	<p>Desarrollador: Hace preguntas basadas en su comprensión del sistema, al tiempo que escribe comportamientos adicionales necesarios desde una perspectiva de desarrollo.</p>	<p>Diseñador Principal: Produce la Descripción Arquitectónica. Se supone que debe ser al menos un profesional de Nivel 3. En Metodologías Ágiles se definen tres niveles de experiencia: Nivel 1 es capaz de “seguir los procedimientos”. Nivel 2 es capaz de “apartarse de los procedimientos específicos” y encontrar otros distintos. Nivel 3 es capaz de manejar con lucidez, mezclar e inventar procedimientos. El Diseñador Principal tiene roles de coordinador, arquitecto, mentor y programador más experto.</p> <p>Diseñador Programador: Produce, junto con el Diseñador Principal, los Borradores de Pantallas, el Modelo Común de Dominio, las Notas y Diagramas de Diseño, el Código Fuente, el Código de Migración, las Pruebas y el Sistema Empaquetado. ¡Un programa en CC es “diseño y programa”; sus programadores son diseñadores programadores. En CC un diseñador que no programe no tiene cabida.</p> <p>Verificador: Produce el Reporte de Bugs. Puede ser un programador en tiempo parcial, o un equipo de varias personas.</p> <p>Escritor: Produce el Manual de Usuario. El Equipo como Grupo es responsable de producir la Estructura y Convenciones del Equipo y los Resultados del Taller de Reflexión.</p>
Equipos	Especializado	Especializado o Multifuncional	Especializado o Multifuncional	Especializado o Multifuncional	Especializado
Prácticas / Reglas	12	7	6	4	7
Limitación Work In Progress	Limitación por iteración	Limitación por Pruebas	Limitación por Pruebas	Limitación por "escenario" y "especificación"	Limitación por Entregables
Incorporación de Tareas	No es posible hasta terminar la iteración	No es posible hasta terminar el requisito actual	No es posible hasta aceptar la prueba	No es posible hasta Refactorizar la Historia de Usuario	No es posible se termina un entregable para empezar con la siguiente iteración
Seguimiento de tareas	<p>Velocity: Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.</p>	<p>Se asegura de que todo el código escrito está cubierto por una prueba. Esto puede dar al programador un mayor nivel de confianza en el código. Sino se encuentra error en las pruebas se procede a refactorizar y terminar.</p>	<p>Los equipos que siguen el proceso de prueba de aceptación han experimentado aumentos de eficiencia. Debido a que el cliente comercial, el desarrollador y el probador están involucrados en la creación de la prueba de aceptación, hay una integración más estrecha entre los equipos multifuncionales. Además, aprobar las pruebas de aceptación demuestra visiblemente que la historia está completa.</p>	<p>Stand Up diario: Este nivel de planificación se lleva a cabo dentro de una sesión diaria que no dura más de 15 minutos, donde cada miembro del equipo responde las siguientes 3 preguntas: ¿Qué logré ayer? ¿Qué planeo lograr hoy? ¿Alguna impedimenta en mi camino?</p>	<p>Frequent Delivery: Consiste en liberar código ejecutable y testeado a usuarios reales cada pocas semanas o meses. Al hacer esto aseguramos que nuestro proyecto no se despegara mucho del cliente necesita realmente, pues cada entrega formal, resultara en correcciones a los problemas que surgen constantemente, lo cual garantiza que se avanza con el desarrollo de manera fiable y rápida.</p>
Estimación	Obligatoria	Opcional	Opcional	Opcional	Opcional

Tabla 1: Comparativo de Metodologías Ágiles
Fuente: Creación Propia a través de la información de diversas fuentes

b. PROGRAMACIÓN EXTREMA (XP)

De acuerdo con el análisis se ha determinado; que la metodología que se adecua a nuestro proyecto y nuestra forma de trabajo es la programación extrema (XP).

Ya que la metodología ágil XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Castillo , Figueroa , & Sevilla , 2019)

i. FASES DE LA METODOLOGÍA XP

a) 1ª FASE: PLANIFICACIÓN DEL PROYECTO

Los proyectos bajo esta metodología se apertura con las historias de usuarios las cuales son determinadas directamente con el cliente, la cual nos permite evaluar nuestros tiempos y pruebas cuando llegue el momento, seguimos con el plan de publicaciones, acá se define la prioridad de las historias de usuarios y los tiempos en los que serán presentadas, debiendo cumplir con el objetivo de cada historia; luego veremos las iteraciones, y según XP se divide en 3 semanas de duración aproximadamente , las cuales deben ser coordinadas con el cliente. En esta fase estimaremos la rapidez del proyecto, la cual se da de acuerdo con las historias de usuarios que se realizarán en cada iteración; se propone la programación en pareja ya que incrementa la productividad en el proyecto.

Se pide también las reuniones diarias con el cliente, consiguiendo así su feedback⁸ sobre los avances.

⁸ Feedback: Significa retroalimentación

b) 2ªFASE: DISEÑO

En la metodología XP, se indica que los diseños deben ser simples, sencillos e intuitivos, para el fácil entendimiento y reutilización de código del proyecto se debe contar con un glosario de términos; se debe pensar los riesgos que pueden surgir en la ejecución del proyecto, por lo cual se sugiere que una pareja de desarrolladores investiguen y reduzcan al máximo los riesgos, para esto nunca se debe añadir funcionalidad extra al programa, aunque se piense que en un futuro será utilizada. Se debe considerar que refactorizar es mejorar y modificar la estructura y codificación de códigos ya establecidos sin perturbar su funcionalidad para evitar malos hábitos se recomienda usar las tarjetas C.R.C⁹.

c) 3ª FASE: CODIFICACIÓN

La codificación se debe estar orientada a la atención de estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad. Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear estas pruebas antes nos ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y sabremos que una vez implementado pasará dichas pruebas sin problemas ya que dicho código ha sido diseñado para ese fin.

d) 4ª FASE: PRUEBAS

Uno de los pilares de la metodología XP, es el uso de un test para comprobar el funcionamiento de los códigos que se vayan implementando, se tiene el test de aceptación que los cuales son creados para verificar la satisfacción y usabilidad del código desarrollado con el cliente, el cual corroborará si cumple con el objetivo de las historias de usuarios que se indicó al inicio.

⁹ En inglés Class, Responsibilities and Collaboration

ii. LAS HISTORIAS DE USUARIO XP

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas.

Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Respecto de la información contenida en la historia de usuario, existen varias plantillas sugeridas, pero no existe un consenso al respecto.

En muchos casos se propone utilizar un nombre y una descripción o sólo una descripción, más quizás una estimación de esfuerzo en días. Si se tienen menos, probablemente sea conveniente dividir las historias, si se tienen más lo mejor es disminuir el detalle y agruparlas. Para efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración). No hay que preocuparse si en un principio no se identifican todas las historias de usuario.

Al comienzo de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración.

Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

iii. ROLES XP

Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado describiremos los roles de acuerdo con la propuesta original de Beck, que son los siguientes:

a) PROGRAMADOR

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

b) CLIENTE

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es solo uno dentro del proyecto, pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

c) ENCARGADO DE PRUEBAS (TESTER)

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

d) ENCARGADO DE SEGUIMIENTO (TRACKER)

El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables

con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

e) ENTRENADOR (COACH)

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

f) CONSULTOR

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

g) GESTOR (BIG BOSS)

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

iv. PRÁCTICAS XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que describiremos a continuación:

a) EL JUEGO DE LA PLANIFICACIÓN

Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. Esta práctica se puede ilustrar como un juego, donde existen dos tipos de jugadores: Cliente

y Programador. El cliente establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes. Este juego se realiza durante la planificación de la entrega, en la planificación de cada iteración y cuando sea necesario reconducir el proyecto.

b) ENTREGAS PEQUEÑAS

La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema, pero sí que constituyan un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

c) METÁFORA

En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. Martin Fowler explica que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema.

d) DISEÑO SIMPLE

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser

removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos.

e) PRUEBAS

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

f) REFACTORIZACIÓN (REFACTORING)

La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo. Robert Martin señala que el diseño del sistema de software es una cosa viviente. No se puede imponer todo en un inicio, pero en el transcurso del tiempo este diseño evoluciona conforme cambia la funcionalidad del sistema. Para mantener un diseño apropiado, es necesario realizar actividades de cuidado continuo durante el ciclo de vida del proyecto. De hecho, este cuidado continuo sobre el diseño es incluso más importante que el diseño inicial. Un concepto pobre al inicio puede ser corregido con esta actividad continua, pero sin ella, un buen diseño inicial se degradará.

g) PROGRAMACIÓN EN PAREJAS

Toda la producción de código debe realizarse con trabajo en parejas de programadores. Según Cockburn y Williams en un estudio realizado para identificar los costos y beneficios

de la programación en parejas, las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan más su trabajo. Dichos beneficios se consiguen después de varios meses de practicar la programación en parejas. En los estudios realizados por Cockburn y Williams este lapso varía de 3 a 4 meses.

h) PROPIEDAD COLECTIVA DEL CÓDIGO

Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

i) INTEGRACIÓN CONTINUA

Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido. Martin Fowler afirma que el desarrollo de un proceso disciplinado y automatizado es esencial para un proyecto controlado, el equipo de desarrollo está más

preparado para modificar el código cuando sea necesario, debido a la confianza en la identificación y corrección de los errores de integración.

j) 40 HORAS POR SEMANA

Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo. Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen al final ser entregados con retraso. En lugar de esto se puede realizar el juego de la planificación para cambiar el ámbito del proyecto o la fecha de entrega.

k) CLIENTE IN-SITU

El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada. Jeffries indica que se debe pagar un precio por perder la oportunidad de un cliente con alta disponibilidad. Algunas recomendaciones propuestas para dicha situación son las siguientes: intentar conseguir un representante que pueda estar siempre disponible y que actúe como interlocutor del cliente, contar con el cliente al menos en las reuniones de planificación, establecer visitas frecuentes de los programadores al cliente para validar el sistema, anticiparse a los problemas asociados estableciendo llamadas telefónicas frecuentes y conferencias, reforzando el compromiso de trabajo en equipo.

l) ESTÁNDARES DE PROGRAMACIÓN

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

m) COMENTARIOS RESPECTO DE LAS PRÁCTICAS

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras. Esto se ilustra en la Figura 1 (obtenida de [2]), donde una línea entre dos prácticas significa que las dos prácticas se refuerzan entre sí. La mayoría de las prácticas propuestas por XP no son novedosas, sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

4. HERRAMIENTAS

Las herramientas que hemos usado para esta tesis están clasificadas de la siguiente manera, considerar que tenemos dos ámbitos el cual es web y móvil:

a. SISTEMA WEB

i. INTERFAZ WEB

Para lograr una interfaz intuitiva, agradable a la vista del usuario y muy práctica, hemos usado las siguientes tecnologías:

1. PYTHON

Es un lenguaje de programación multiparadigma ya que es orientado a objetos, es una programación imperativa y en cierta forma es programación funcional, de código claramente legible y potente, comparable a la de Perl, Ruby, Scheme, o Java. Python combina, potencia con una sintaxis muy clara. Tiene módulos, clases, excepciones, débilmente tipado. Hay muchas interfaces para las llamadas al sistema y bibliotecas, así como a varios sistemas de ventanas. Los nuevos módulos incorporados se escriben fácilmente en C o C ++ (u otros idiomas, dependiendo de la aplicación elegida).

“En un texto considerado de dominio público llamado EL ZEN DE PYTHON indica los siguientes principios de este lenguaje:

- Bonito es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Espaciado es mejor que denso.
- La legibilidad es importante.
- Los casos especiales no son lo suficientemente especiales como para romper las reglas.
- Sin embargo, la practicidad le gana a la pureza.
- Los errores nunca deberían pasar silenciosamente.
- A menos que se silencien explícitamente: Frente a la ambigüedad, evita la tentación de adivinar.
- Debería haber una, y solamente una, manera obvia de hacerlo. A pesar de que no sea obvio a menos que seas holandés.

- Ahora es mejor que nunca: A pesar de que nunca es muchas veces mejor que justo ahora.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, quizás sea una buena idea.

Los espacios de nombres son una gran idea, ¡tenemos más de esas!” (Peters, 1999)

a) CARACTERÍSTICAS NOTABLES DE PYTHON

- Utiliza una sintaxis elegante, por lo que los programas se escriben más fácil de leer.
- Es un idioma fácil de usar, hace que sea sencillo para obtener su trabajo del programa.
Esto hace Python ideal para el desarrollo de prototipos y otras tareas de programación AD-HOC, sin comprometer la capacidad de mantenimiento.
- Se puede ampliar fácilmente añadiendo nuevos módulos implementados en un lenguaje compilado como C o C ++ también puede ser embebido en una aplicación para proporcionar una interfaz programable.
- Se ejecuta en cualquier lugar, incluyendo Mac OS X, de Windows, Linux y Unix.
- Es software libre en dos sentidos. No cuesta nada para descargar o utilizar Python, o incluirlo en su aplicación.

b) ¿QUÉ HACE PYTHON EN NUESTRO SISTEMA?

- Conectarse a la base de datos (persistencia de datos).
- Es el Back End: Cuando nos referimos a este término Back – End, encontramos que es el responsable en una programación web de todos los componentes dinámicos, se dedica a la parte lógica del sitio web, en pocas palabras es el responsable que todo funcione de la mejor manera, optimiza recursos y brinda seguridad al sitio web.

- Los lenguajes de programación que se utilizan en el Back-End en la actualidad son PHP, JAVASCRIPT, PHYTON Y RUBY. También se requiere de conocimientos en de HTML y CSS, aunque a un nivel básico.
- Las herramientas que se utilizan en el Back-End son editores de código, compiladores, algunos debuggeadores para revisar errores y seguridad, gestores de bases de datos y algunas otras cosas.
- Permite realizar los cálculos del dictamen del aviso (Indemnizable y No indemnizable).
- Realiza los Cálculos del punto de evaluación, usando las librerías OSGEO y MAPSCRIPT

2. DJANGO:

Es un framework para aplicaciones web open source y gratuito, está escrito en Python de alto nivel que fomenta el rápido desarrollo y diseño limpio y pragmático. Es de código libre y abierto, respeta el patrón de diseño conocido como Modelo–vista–controlador.

Django fue diseñado para ayudar a los desarrolladores a llevar las aplicaciones desde el concepto hasta su finalización lo más rápido posible, ayuda a los desarrolladores a evitar muchos errores comunes de seguridad, como la inyección de SQL, scripts entre sitios, falsificación de solicitudes entre sitios y clickjacking. Su sistema de autenticación de usuarios proporciona una forma segura de administrar cuentas de usuario y contraseñas. Algunos de los sitios más concurridos de la Web aprovechan la capacidad de Django para escalar de forma rápida y flexible. (Holovaty & Kaplan , 2009)

Con Django, puede trasladar las aplicaciones web desde el concepto hasta el lanzamiento en cuestión de horas. Django se encarga de gran parte de las molestias del desarrollo web, por

lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto.

Django se encarga de la autenticación de usuarios, la administración de contenidos, los mapas del sitio, los canales RSS y muchas más tareas, nada más sacarlo de la caja.

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un framework MVC. Someramente, la M, V y C se separan en Django de la siguiente manera:

- M, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django, la cual describiremos en este capítulo.
- V, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas.
- C, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo siguiendo tu URL con f y llamando a la función apropiada de Python para la URL obtenida.

Debido a que la "C" es manejada por el mismo framework y la parte más importante se produce en los modelos, las plantillas y las vistas, Django es conocido como un Framework MTV. En el patrón de diseño MTV indica lo siguiente:

- M significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.

- V significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelo y las plantillas.

a) CARACTERÍSTICAS NOTABLES DE DJANGO

- Un mapeado objeto relacional.
- Una API de base de datos robusta.
- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- Un despachador de URL basado en expresiones regulares.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.

b) ¿QUÉ HACE DJANGO EN NUESTRO SISTEMA?

Nos ayuda en el Modelo en la vista y en el template de nuestro sistema.

3. BOOT STRAP

Es un kit de herramientas de código abierto para diseño de sitios y aplicaciones web las cuales están desarrolladas con HTML, CSS y JS. Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. Una hoja de estilo llamada bootstrap.less incluye los componentes de las hojas de estilo. (Bootstrap, 2019)

Bootstrap proporciona un conjunto de hojas de estilo que proveen definiciones básicas de estilo para todos los componentes de HTML. Esto otorga una uniformidad al navegador y al sistema de anchura, da una apariencia moderna para el formateo de los elementos de texto, tablas y formularios. (Bootstrap, 2019)

Los componentes de JavaScript para Bootstrap están basados en la librería jQuery de JavaScript. Los plugins se encuentran en la herramienta de plugin de jQuery. Proveen

elementos adicionales de interfaz de usuario como diálogos, tooltips y carruseles. También extienden la funcionalidad de algunos elementos de interfaz existentes, incluyendo por ejemplo una función de autocompletar para campos de entrada (input). La versión 2.0 soporta los siguientes plugins de JavaScript: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel y Typeahead.

a) CARACTERÍSTICAS NOTABLES DE BOOTSTRAP

- Compatible con la mayoría de los navegadores web.
- Código abierto y está disponible en GitHub.
- Viene con una disposición de cuadrilla estándar de 940 píxeles de ancho.
- Otorga una uniformidad al navegador y al sistema de anchura, da una apariencia moderna para el formateo de los elementos.

4. JAVASCRIPT

Es un lenguaje de programación de scripts (secuencia de comandos). JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. JavaScript es regularmente utilizado para hacer extensiones para diferentes programas, como los scripts codificados en LUA O PYTHON. (Gutierrez, 2009)

a) CARACTERÍSTICAS NOTABLES DE JAVASCRIPT

- JavaScript es un lenguaje de programación que necesita un intérprete para ser ejecutado.
- JavaScript se utiliza principalmente en páginas web.

- Es Imperativo y estructurado, dinámico y funcional

5. JQUERY

Es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Hace cosas como el recorrido y manipulación de documentos HTML, manejo de eventos, animación, y Ajax mucho más simple con una API fácil de usar que funciona en una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, gracias al jQuery ha cambiado la forma en que millones de personas escriben JavaScript.

jQuery nos permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

a) CARACTERÍSTICAS NOTABLES DE JQUERY

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.

6. HTML

Sus siglas en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit en GNU/Linux, el Bloc de notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, Tex-tPad, Vim, Notepad++, entre otros.

a) CARACTERÍSTICAS NOTABLES DE HTML

- Facilita el desarrollo de páginas web compatibles con distintos navegadores y plataformas.
- Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

7. CSS

En español son hojas de estilo en cascada, pero se le conoce como CSS por sus siglas en inglés de Cascading Stylesheets Stylesheets.

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Bastante usado para crear el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web, y GUI's para muchas aplicaciones móviles.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes.

Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características para las presentaciones, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

ii. SISTEMA OPERATIVO

Para nuestro sistema hemos utilizados el sistema operativo LINUX en la cual tenemos las siguientes herramientas:

1. MYSQL

Sistema de gestión de base de datos relacional, se puede trabajar con licencia gratuita o comercial. MySQL es muy utilizado en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación. Las siguientes características son implementadas únicamente por MySQL (ORACLE, 2019):

- Permite escoger entre múltiples motores de almacenamiento para cada tabla. En MySQL 5.0 éstos debían añadirse en tiempo de compilación, a partir de MySQL 5.1 se pueden añadir dinámicamente en tiempo de ejecución.
- Los hay nativos como MyISAM, Falcon, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example.
- Desarrollados por partners como solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (antes Brighthouse), Kickfire, XtraDB, IBM DB2. InnoDB Estuvo desarrollado así pero ahora pertenece también a Oracle.
- Desarrollados por la comunidad como memcache, httpd, PBXT y revisión.

- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

2. NGINX

Se pronuncia en inglés “engine X”, es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3), software libre de código abierto. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows.

a) CARACTERÍSTICAS DEL SERVIDOR WEB NGINX

- Servidor de archivos estáticos, índices y autoindexado.
- Proxy inverso con opciones de caché.
- Balanceo de carga.
- Tolerancia a fallos.
- Soporte de HTTP y HTTP2 sobre SSL.
- Soporte para FastCGI con opciones de caché.
- Servidores virtuales basados en nombre y/o en dirección IP.
- Streaming de archivos FLV y MP4.14.
- Soporte para autenticación.
- Compatible con IPv6.
- Soporte para protocolo SPDY.
- Compresión gzip.
- Habilitado para soportar más de 10.000 conexiones simultáneas.

b) CARACTERÍSTICAS DEL PROXI WEB

- Proxy SMTP, POP3 e IMAP.
- Soporta STARTTLS.

- Soporta SSL.

3. UWSGI

Tiene como objetivo desarrollar una pila completa para construir servicios de alojamiento. Los servidores de aplicaciones (para varios lenguajes de programación y protocolos), proxis, administradores de procesos y monitores se implementan utilizando una API común y un estilo de configuración común. En otras palabras, es un servidor de contenedor de aplicaciones rápido, auto regenerativo y amigable para desarrolladores.

Es un software de código abierto BSD, para el módulo de integración de NGINX, GPL para el módulo de integración de apache2. WSGI Entrega una pila completa para aplicaciones WEB conectadas en red/cluster, implementación mensajes/paso de objetos, cache, RPC y comunicación entre procesos. Se puede ejecutar en los modos de preforking, threaded, asynchronous/eventos y soporta diversas formas de hilos tales (uGreen, Greenlet, Fiber), Ofrece varios métodos de configuración: a través de línea de comandos, variables de entorno, otra vez de XML, INI, archivos YAML, a través de LDAP y muchos más.

4. VIRTUAL ENV

Es una herramienta para crear entornos de Python aislados. Posee un directorio para ubicar el nuevo entorno virtual. Tiene una serie de efectos habituales (modificables por muchas opciones):

- ENV/lib/ y ENV/include/ se crean, que contienen archivos de biblioteca de apoyo para una nueva python virtualenv.
- Los paquetes instalados en este entorno vivirán bajo ENV/lib/pythonX.X/site-packages/.

- ENV/bin se crea, donde los ejecutables viven, notablemente un nuevo Python. Por lo tanto, al ejecutar un script se ejecutaría ese script bajo el Python de este virtualenv.#!/path/to/ENV/bin/python
- Se instalan las herramientas pip y setuptools , que permiten que otros paquetes se instalen fácilmente en el entorno. Este pip asociado se puede ejecutar desde ENV/bin/pip.
- El Python en su nuevo virtualenv está efectivamente aislado del Python que se usó para crearlo.

5. GIT

Es un sistema de control de versiones distribuidas de código abierto y gratuito diseñado para manejar todo, desde proyectos pequeños a muy grandes, con velocidad y eficiencia.

Git es fácil de aprender y tiene una huella pequeña con un rendimiento increíblemente rápido.

Supera a las herramientas de SCM como Subversión, CVS, PERFORCE Y CLEARCASE con funciones como ramificación local barata, áreas de preparación conveniente y flujos de trabajo múltiples.

La característica Git que realmente la distingue de casi todas las demás SCM que existe es su modelo de ramificación.

Git te permite y te anima a tener múltiples sucursales locales que pueden ser totalmente independientes entre sí. La creación, fusión y eliminación de esas líneas de desarrollo lleva segundos.

a) GIT EN NUESTRO SISTEMA

- **Conmutación de contexto sin fricción:** Permite crear un ambiente para probar una idea, comprométase un par de veces, vuelva al lugar desde el que se ramificó, aplique un parche, vuelva al lugar donde está experimentando y combínelo.
- **Codelines basadas en roles:** Disponga de una sucursal que siempre contenga solo lo que se destina a la producción, otra en la que combine trabajo para realizar pruebas y otra más pequeña para el trabajo diario.
- **Flujo de trabajo basado en funciones:** Cree nuevas sucursales para cada nueva función en la que esté trabajando, de modo que pueda alternar sin problemas entre ellas, luego elimine cada una de ellas cuando esa característica se fusione en su línea principal.
- **Experimentación desechable:** Cree una rama para experimentar, descubra que no va a funcionar, y simplemente elimínela (abandonando el trabajo) sin que nadie más la haya visto (incluso si ha empujado otras ramas mientras tanto).

5. DESCRIPCIÓN GRÁFICA DE CÓMO TRABAJAN ESTAS HERRAMIENTAS

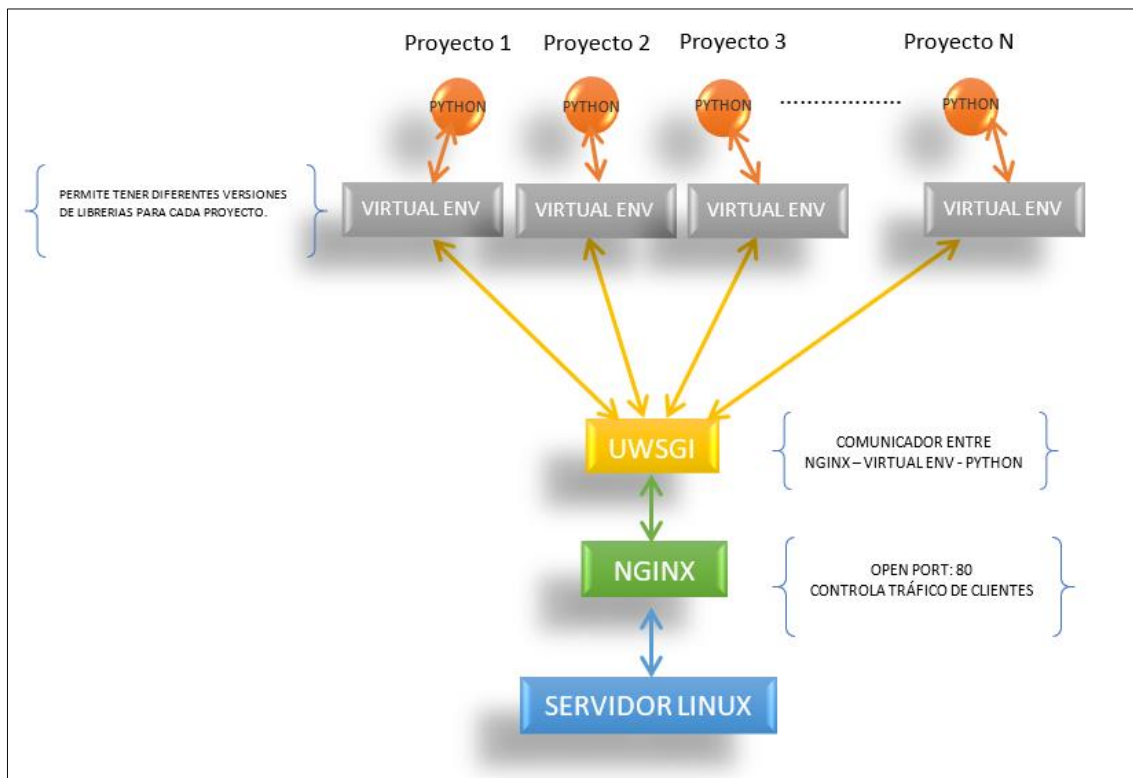


Imagen 5: Arquitectura a nivel de Software del Sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico Lima – 2017
Fuente: Creación propia

CAPÍTULO III: RESULTADOS Y DISCUSIÓN

1. METODOLOGÍA XP

I. PLANIFICACIÓN DEL PROYECTO

A. REQUERIMIENTOS FUNCIONALES:

i. ITERACIÓN I

1. HISTORIA DE USUARIO N°1: INTEGRAR CARTOGRAFÍA CON SECTORES ESTADÍSTICOS.

HISTORIA DE USUARIO			
IDENTIFICADOR:	01		
NOMBRE:	Integrar cartografía con sectores estadísticos.		
PRIORIDAD:	Alta	USUARIO:	MINAGRI
ESTIMACIÓN:	14	ITERACIÓN:	01
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo:		Como	MINAGRI
Quiero: Que es sistema genere la cartografía completa de los sectores estadísticos.			
Para: Su respectiva evaluación y verificación.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar cartografía del usuario		4	
Diseñar estructuras para la base de datos		1	
Crear base de datos e integrar los sectores asegurados con la póliza enviada por la aseguradora (Distritos, sector estadístico, cultivos y tipo de evento asegurados por campaña).		5	
Programar base de datos		2	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Todos los sectores tienen cartografía			
Dado: Que ya se tiene en la base de datos la cartografía			
Cuando: El Representante de región registre un aviso			
Entonces: Debe mostrar la cartografía del sector estadístico.			

Tabla 2: HU N°1: Integrar cartografía con sectores estadísticos.
Fuente: Creación Propia a través de la información brindada por el cliente

Como se muestra en la imagen 6 es el prototipo de la HU N1, cabe mencionar que esta imagen fue brindada por el cliente:

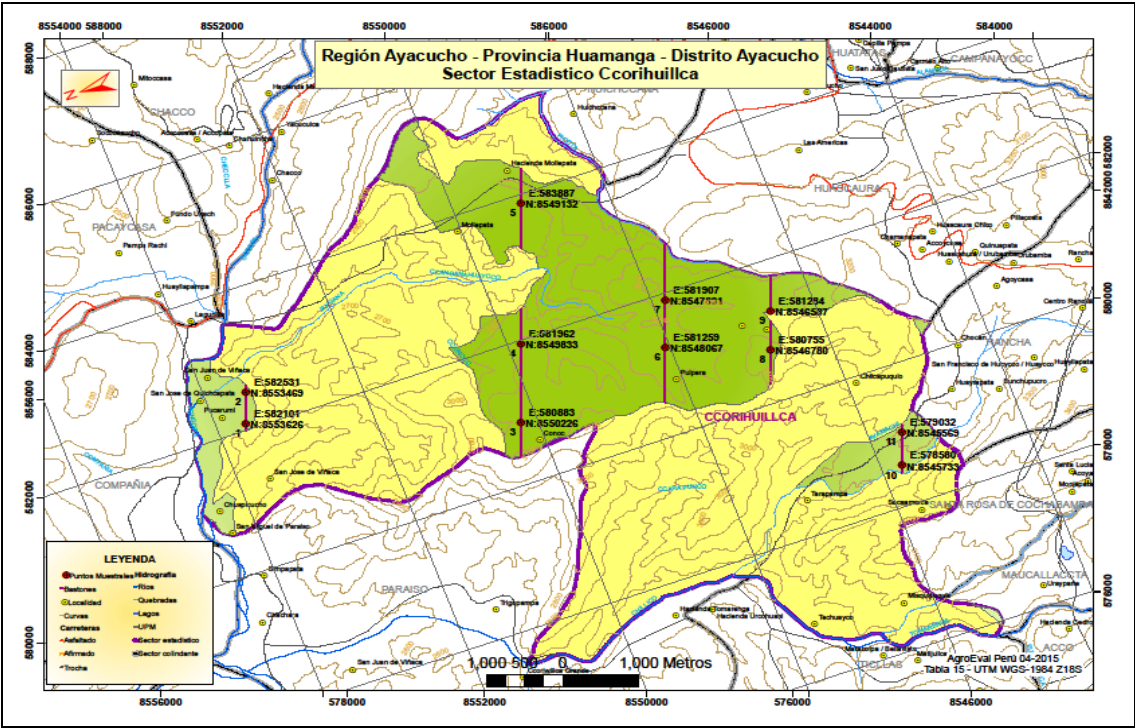


Imagen 6: Prototipo Integrar cartografía con sectores estadísticos.
Fuente: Imagen proporcionada por el cliente

ii. ITERACIÓN II

1. HISTORIA DE USUARIO N°2: REGISTRAR AVISO

HISTORIA DE USUARIO			
IDENTIFICADOR:	02		
NOMBRE:	Registrar aviso		
PRIORIDAD:	Alta	USUARIO:	Representante de Región
ESTIMACIÓN:	8	ITERACIÓN:	02
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Como representante de región			
Quiero: Ingresar todos los datos del siniestro ocurrido.			
Para: Su respectiva evaluación y verificación.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		4	

Realizar pruebas y corregir errores	1
CRITERIOS DE ACEPTACIÓN	
Escenario 1: Al ingresar los avisos según póliza	
Dado: que el representante de región necesita registrar un aviso Cuando: Ingrese los datos del aviso Entonces: El sistema mostrará un mensaje de “Aviso registrado con éxito”.	
Escenario 2: Al ingresar los avisos que no están en póliza	
Dado: que el representante de región necesita registrar un aviso Cuando: Ingrese los datos del aviso. Entonces: El sistema sólo mostrará en los combos de selección, los datos de póliza.	
Escenario 3: No se puede Volver a calcular los puntos	
Dado: que el representante de región necesita cambiar los puntos de evaluación que genera el sistema Cuando: el Represente de región termine de ingrese los datos del aviso y se genere el mapa Entonces: El sistema mostrará un mensaje de error: “Los puntos no fueron cambiados”	

Tabla 3: HU N°2: Registrar aviso
Fuente: Creación Propia a través de la información brindada por el cliente

El prototipo de la HU2 se muestra en la imagen 7, la cual es el siguiente:

Registrar Aviso

Departamento:

Provincia:

Sector estadístico:

Cultivo:

Tipo de Evento:

Imagen 7: Prototipo Registrar aviso
Fuente: Creación propia

2. HISTORIA DE USUARIO N° 3: GENERAR MAPA DEL SECTOR ESTADÍSTICO CON PUNTOS A EVALUAR – WEB

HISTORIA DE USUARIO			
IDENTIFICADOR:	03		
NOMBRE:	Generar Mapa del Sector Estadístico con puntos a evaluar - Web		
PRIORIDAD:	Alta	USUARIO:	Representantes de región
ESTIMACIÓN:	8	ITERACIÓN:	02
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
<p>Cómo: Como representante de Región</p> <p>Quiero: Generar el mapa del sector estadístico que se va a evaluar</p> <p>Para: Identificar geográficamente al sector estadístico que se va a evaluar.</p>			
OBSERVACIONES			
Los puntos de evaluación deben ser generados por el sistema.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		4	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Generar mapa por aviso			
<p>Dado: A que el Representante de región registra avisos</p> <p>Cuando: el representante de región ingrese los datos de ubicación del siniestro (Departamento, Provincia, Distrito y Sector Estadístico)</p> <p>Entonces: El sistema mostrará el mapa de dicho sector estadístico.</p>			
Escenario 2: No muestra mapa del sector estadístico			
<p>Dado: A que el Representante de región registra avisos</p> <p>Cuando: el representante de región ingrese los datos de ubicación del siniestro (Departamento, Provincia, Distrito y Sector Estadístico)</p> <p>Entonces: El sistema mostrará una imagen que indica CARTOGRAFÍA NO PROPORCIONADA</p>			

Tabla 4:HU N°3: Generar Mapa del Sector Estadístico con puntos a evaluar – Web

Fuente: Creación Propia a través de la información brindada por el cliente

El prototipo de la HU3 se muestra en la imagen 8, la cual es el siguiente:

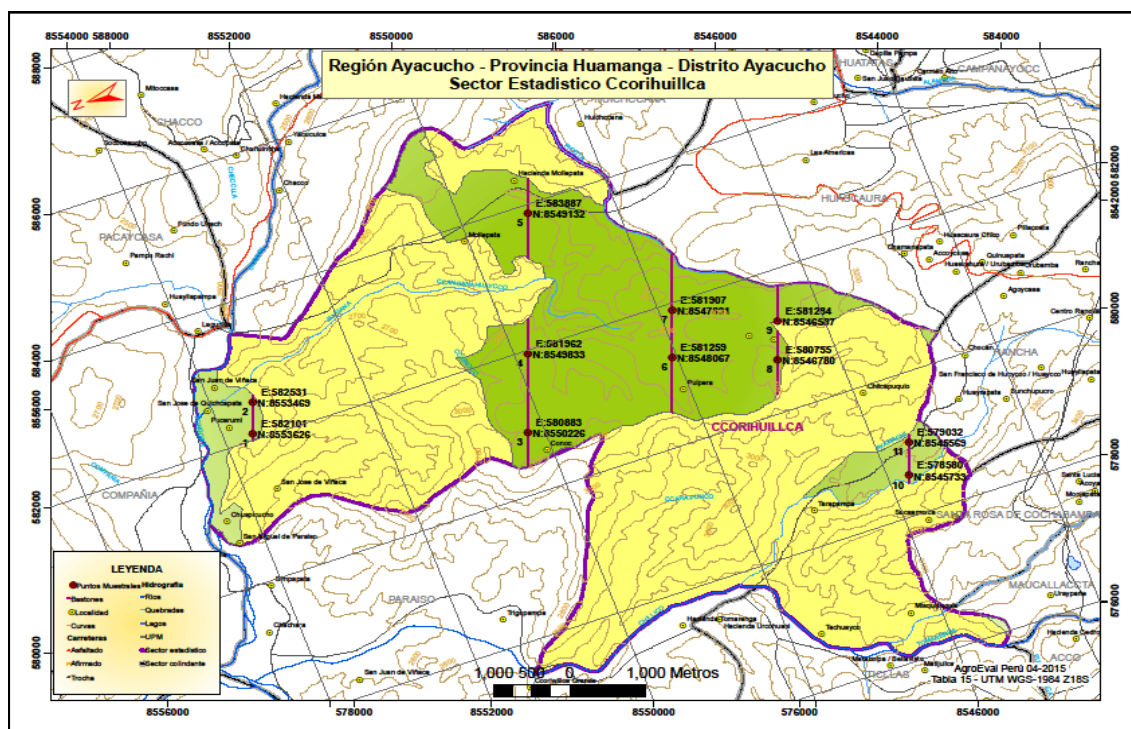


Imagen 8: Prototipo Generar Mapa del Sector Estadístico con puntos a evaluar – Web
Fuente: Imagen proporcionada por el cliente

3. HISTORIA DE USUARIO N°4: MOSTRAR MAPA DEL SECTOR ESTADÍSTICO – TABLET

HISTORIA DE USUARIO			
IDENTIFICADOR:	04		
NOMBRE:	Mostrar Mapa del Sector Estadístico - Tablet		
PRIORIDAD:	Alta	USUARIO:	Evaluador
ESTIMACIÓN:	8	ITERACIÓN:	02
DESARROLLADORES:	<ul style="list-style-type: none">• Cinthya Amaya• Christian Rodas		
DESCRIPCIÓN			
Cómo:		Como	Evaluador
Quiero: Visualizar el mapa del sector estadístico a evaluar con sus respectivos puntos de evaluación.			
Para: Identificar geográficamente al sector estadístico que se va a evaluar.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		4	
Realizar pruebas y corregir errores		1	

CRITERIOS DE ACEPTACIÓN
Escenario 1: Sí muestra cartografía por aviso Dado: a que se requiere visualizar los mapas por aviso Cuando: El evaluador ingrese a sus avisos que ha descargado en la Tablet Entonces: El sistema debe mostrar los mapas del sector estadístico por aviso.
Escenario 2: No muestra cartografía por aviso Dado: a que se requiere visualizar los mapas por aviso Cuando: El evaluador ingrese a sus avisos que ha descargado en la Tablet Entonces: El sistema debe mostrar una imagen en lugar del mapa indicando CARTOGRAFÍA NO PROPORCIONADA

Tabla 5: HU N° 4: Mostrar Mapa del Sector Estadístico – Tablet
Fuente: Creación Propia a través de la información brindada por el cliente

El prototipo de la HU4 se muestra en la imagen 9, la cual es el siguiente:

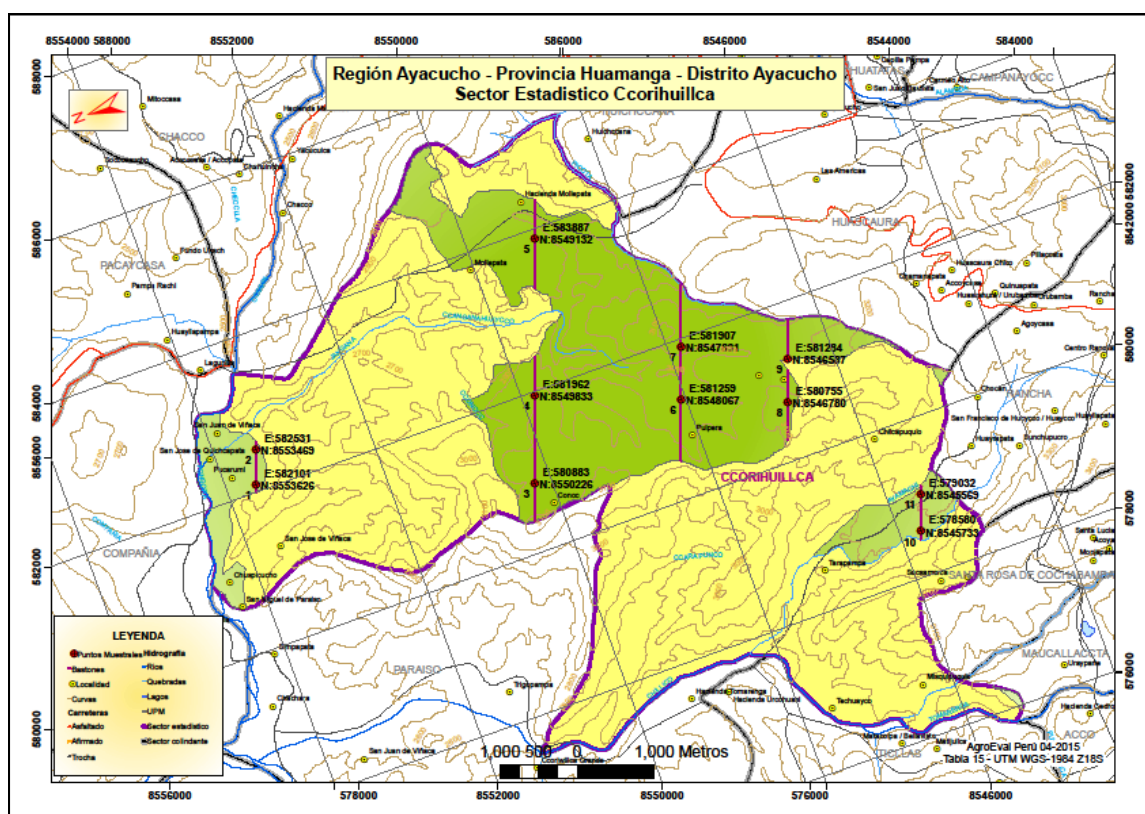


Imagen 9: Prototipo Mostrar Mapa del Sector Estadístico – Tablet
Fuente: Imagen proporcionada por el cliente

4. HISTORIA DE USUARIO N°5: ADMINISTRAR BANDEJA DE AVISOS

HISTORIA DE USUARIO			
IDENTIFICADOR:	05		
NOMBRE:	Administrar Bandeja de Avisos		
PRIORIDAD:	Alta	USUARIO:	Evaluador
ESTIMACIÓN:	5	ITERACIÓN:	02
DESARROLLADORES:	<ul style="list-style-type: none">• Cinthya Amaya• Christian Rodas		
DESCRIPCIÓN			
Cómo: Como evaluador. Quiero: visualizar mis avisos según estados PENDIENTES – TERMINADOS Y ENVIADOS. Para: realizar la respectiva evaluación y verificación.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		1	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: ubicación de avisos en las bandejas			
Dado: que son varios avisos Cuando: El evaluador quiera ver sus avisos Entonces: El sistema mostrará los avisos según en la bandeja que se encuentre.			

Tabla 6: HU N° 5: Administrar Bandeja de Avisos
Fuente: Creación Propia a través de la información brindada por el cliente

El prototipo de la HU5 se muestra en la imagen 10, la cual es el siguiente:

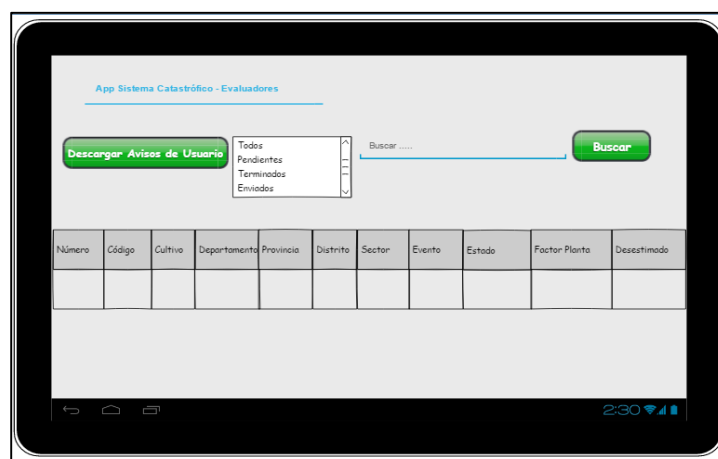


Imagen 10: Prototipo Administrar Bandeja de Avisos
Fuente: Creación propia

5. HISTORIA DE USUARIO N° 6: DESCARGAR AVISOS EN TABLET

HISTORIA DE USUARIO			
IDENTIFICADOR:	06		
NOMBRE:	Descargar Avisos En Tablet		
PRIORIDAD:	Alta	USUARIO:	Evaluador
ESTIMACIÓN:	5	ITERACIÓN:	02
DESARROLLADORES:	<ul style="list-style-type: none">Cintha AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Como evaluador. Quiero: descargar todos los avisos que están pendientes para evaluación. Para: realizar la respectiva evaluación y verificación.			
OBSERVACIONES			
Las descargas de los avisos se deben realizar cuando el dispositivo está conectado a internet.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		1	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Descargar avisos			
Dado: que se tiene que descargar los avisos que el representante de región ha enviado			
Cuando: el evaluador solicite descargar sus avisos			
Entonces: el sistema mostrara un mensaje que indica que la descarga fue exitosa.			
Escenario 2: No hay avisos nuevos por Descargar			
Dado: que se tiene que descargar los avisos que el representante de región ha enviado			
Cuando: el evaluador solicite descargar sus avisos			
Entonces: el sistema mostrara el mensaie “No tiene nuevos avisos por descargar”			

Tabla 7: HU N° 6: Descargar Avisos En Tablet

Fuente: Creación Propia a través de la información brindada por el cliente

El prototipo de la HU6 se muestra en la imagen 11, la cual es el siguiente:

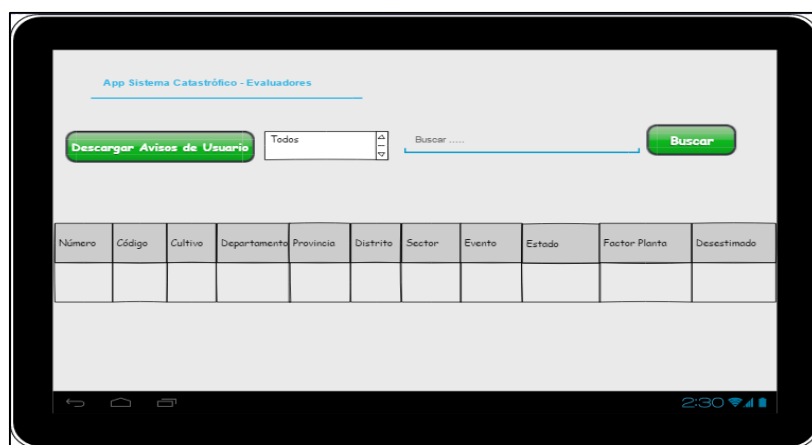


Imagen 11: Prototipo Descargar Avisos En Tablet

Fuente: Creación propia

iii. ITERACIÓN III

1. HISTORIA DE USUARIO N° 7: REGISTRAR DATOS DE SINIESTROS POR AVISO – TABLET

HISTORIA DE USUARIO			
IDENTIFICADOR:	07		
NOMBRE:	Registrar datos de siniestros por aviso - Tablet		
PRIORIDAD:	Alta	USUARIO:	Evaluadores
ESTIMACIÓN:	6	ITERACIÓN:	03
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Como evaluador			
Quiero: Registrar los datos de la evaluación de los siniestros en la Tablet.			
Para: Tener la información de la evaluación y enviársela a la aseguradora y al MINAGRI.			
OBSERVACIONES			
Una vez registrados los datos de los avisos, estos se pueden enviar a la bandeja de terminados.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		2	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Ingreso de datos			
Dado: que se ingresan los datos de las evaluaciones			
Cuando: los evaluadores activen el chek de requiere análisis.			
Entonces: El sistema activará la opción para ingresar los datos como son número de plantas y cultivos por plantas.			
Escenario 2: Se generan los datos automáticamente			
Dado: que se ingresan los datos de las evaluaciones			
Cuando: los evaluadores ingresan los datos solicitados.			
Entonces: El sistema generará los cálculos automáticos de Cultivos por análisis y siembra.			
Escenario 3: No se generan los datos automáticamente			
Dado: que se ingresan los datos de las evaluaciones			
Cuando: los evaluadores ingresan los datos solicitados.			
Entonces: El sistema verificará si los datos ingresados son correctos de lo contrario mostrará el mensaje “El dato ingresado es incorrecto, no se puede generar valores”			
Escenario 4: Se requiere agregar foto por punto			

Dado: se debe tomar una foto al cultivo en el punto de evaluación Cuando: los evaluadores registren los datos del punto también se le solicitará una foto Entonces: El sistema guardará la foto tomada por el evaluador dentro de la aplicación.
Escenario 5: No se puede tomar foto
Dado: se debe tomar una foto al cultivo en el punto de evaluación Cuando: los evaluadores procedan a tomar las fotos sin éxito Entonces: el sistema mostrará el mensaje “No se pudo tomar la foto, inténtelo de nuevo”
Escenario 6: Terminar el aviso
Dado: que se ingresan los datos de las evaluaciones Cuando: los evaluadores van a campo y proceden a terminar la evaluación. Entonces: si no hay ningún error de ingreso de datos el sistema mostrará el siguiente mensaje “El aviso se envió a la bandeja de Terminados”.
Escenario 7: No se puede terminar el aviso
Dado: que se ingresan los datos de las evaluaciones Cuando: los evaluadores van a campo y proceden a terminar la evaluación. Entonces: si existe error de ingreso de datos el sistema mostrará el siguiente mensaje “Los datos ingresados no son correctos, corregir lo siguiente: ...”.

Tabla 8: HU N°7: Registrar datos de siniestros por aviso – Tablet
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU7 se muestra en la imagen 12, la cual es el siguiente:

N°	Este	Norte	Req. Análisis	N° Plantas	Cultivos por Planta	Cultivos por Análisis	Siembra	Foto
			<input type="checkbox"/>					FOTO 1
			<input checked="" type="checkbox"/>					FOTO 2
			<input type="checkbox"/>					FOTO 3
			<input type="checkbox"/>					FOTO 4
			<input checked="" type="checkbox"/>					FOTO 5

Imagen 12: Prototipo Registrar Siniestros por aviso – Tablet
Fuente: Creación propia

2. HISTORIA DE USUARIO N° 8: ENVIAR AVISOS TERMINADOS – TABLET

HISTORIA DE USUARIO			
IDENTIFICADOR:	08		
NOMBRE:	Enviar Avisos Terminados - Tablet		
PRIORIDAD:	Alta	USUARIO:	Evaluadores
ESTIMACIÓN:	6	ITERACIÓN:	03
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Como evaluador			
Quiero: Enviar los avisos terminados al servidor			
Para: que la información pueda ser visualizada por MINAGRI – AGROEVAL y puedan ver el dictamen que genera cada aviso.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		2	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Enviar el aviso a la bandeja de enviados			
Dado: que los avisos que se encuentran en la bandeja de terminados ya están aprobados y verificados por el evaluador.			
Cuando: los evaluadores quieran enviar los avisos al servidor			
Entonces: el sistema mostrará “El aviso fue enviado al servidor con éxito”			
Escenario 2: No se puede enviar el aviso a la bandeja de enviados			
Dado: que los avisos que se encuentran en la bandeja de terminados ya están aprobados y verificados por el evaluador.			
Cuando: los evaluadores quieran enviar los avisos al servidor sin tener acceso a internet			
Entonces: el sistema mostrará “El dispositivo móvil no cuenta con internet”			
Escenario 3: No se puede enviar el aviso a la bandeja de enviados por problemas del servidor			
Dado: que los avisos que se encuentran en la bandeja de terminados ya están aprobados y verificados por el evaluador.			
Cuando: los evaluadores quieran enviar los avisos al servidor sin éxito			
Entonces: el sistema mostrará “El servidor presenta inconvenientes, inténtelo en unos instantes”.			
Escenario 4: No puedo modificar los datos			
Dado: que los avisos que se encuentran en la bandeja de terminados			
Cuando: los evaluadores quieran modificar los datos, ya que se dieron cuenta de algún error			
Entonces: el sistema mostrará la opción de retornar a pendientes, el cual les permitirá modificar los datos.			
Escenario 5: Retornar el aviso a bandeja de Pendientes			
Dado: que los avisos que se encuentran en la bandeja de terminados			
Cuando: los evaluadores deben retornar los avisos a pendientes mediante una opción del sistema.			
Entonces: el sistema mostrará el mensaje “EL AVISO FUE ENVIADO A LA BANDEJA DE PENDIENTES”			

Tabla 9: HU N°8: Enviar Avisos Terminados - Tablet:
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU8 se muestra en la imagen 13, la cual es el siguiente:

Imagen 13: Prototipo Enviar Avisos Terminados – Tablet

Fuente: Creación propia

3. HISTORIA DE USUARIO N° 9: AVISOS ENVIADOS

HISTORIA DE USUARIO			
IDENTIFICADOR:	09		
NOMBRE:	Avisos Enviados		
PRIORIDAD:	Alta	USUARIO:	Evaluadores
ESTIMACIÓN:	6	ITERACIÓN:	03
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo:		Como	evaluador
Quiero: visualizar los avisos enviados al servidor			
Para: tener registro de todos los avisos enviados.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		2	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Visualizar los avisos enviados			
Dado: Los avisos ya están enviados al servidor			
Cuando: el evaluador accede a la bandeja de enviados			
Entonces: El sistema mostrará todos los avisos que fueron enviados al servidor.			

Tabla 10: HU N°9: Avisos Enviados

Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU9 se muestra en la imagen 14, la cual es el siguiente:



Imagen 14: Prototipo Avisos enviados
Fuente: Creación propia

4. HISTORIA DE USUARIO N° 10: CERRAR EL AVISO – WEB

HISTORIA DE USUARIO			
IDENTIFICADOR:	10		
NOMBRE:	Cerrar el Aviso - Web		
PRIORIDAD:	Alta	USUARIO:	Evaluadores
ESTIMACIÓN:	7	ITERACIÓN:	03
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Como evaluador			
Quiero: cerrar los avisos enviados al servidor			
Para: que la información pueda ser visualizada por MINAGRI – AGROEVAL y puedan ver el dictamen que genera cada aviso.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		3	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: cerrando avisos			
Dado: que los avisos ya están en el servidor			
Cuando: el evaluador quiera cerrar los avisos, debe entrar al aviso			
Entonces: el sistema mostrará una opción para cerrar el aviso.			
Escenario 2: No se logra cerrar avisos			
Dado: que los avisos ya están en el servidor			
Cuando: el evaluador quiera cerrar los avisos, debe entrar al aviso			

Entonces: el sistema mostrará un mensaje “No se puede cerrar aviso, hay datos mal ingresados”

Escenario 3: No se logra cerrar avisos por falta de internet

Dado: que los avisos ya están en el servidor

Cuando: el evaluador quiera cerrar los avisos, debe entrar al aviso

Entonces: el sistema mostrará un mensaje “NO CUENTA CON INTERNET”

Tabla 11: HU N° 10: Cerrar el aviso – Web

Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU10 se muestra en la imagen 15, la cual es el siguiente:

Imagen 15: Prototipo Cerrar el aviso – Web

Fuente: Creación propia

iv. ITERACIÓN IV

1. HISTORIA DE USUARIO N° 11: ENVIAR AVISO A POSITIVA

HISTORIA DE USUARIO			
IDENTIFICADOR:	011		
NOMBRE:	Enviar Aviso a Positiva		
PRIORIDAD:	Alta	USUARIO:	MINAGRI
ESTIMACIÓN:	5	ITERACIÓN:	04
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Minagri			
Quiero: Enviar los avisos ya corregidos a Positiva.			
Para: Que procedan con la verificación y evaluación.			
OBSERVACIONES			
Los avisos no se podrán enviar solo por dos razones: No hay internet o hay inconvenientes con el servidor.			
TAREAS			
Descripción		Duración (horas)	

Analizar, diseñar y validar interfaz de usuario	1
Diseñar estructura para la base de datos	1
Crear estructuras para base de datos	1
Desarrollar, construir lógica de negocio e interfaz de usuario	1
Realizar pruebas y corregir errores	1
CRITERIOS DE ACEPTACIÓN	
Escenario 1: Aviso correcto	
Dado: que los datos están conformes	
Cuando: MINAGRI desee enviar los avisos a Positiva	
Entonces: el sistema mostrará una opción para que pueda enviar el aviso a Positiva	
Escenario 2: No se puede enviar aviso	
Dado: que los datos están conformes	
Cuando: MINAGRI desee enviar los avisos a Positiva	
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está enviando el aviso.	

Tabla 12: HU N°11: Enviar Aviso a Positiva
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU11 se muestra en la imagen 16, la cual es el siguiente:

Detalle del aviso

Detalle

Departamento: Provincia:

Distrito: Sector Estadístico:

Cultivo: Tipo de Evento:

Estado: **Minagri**

Factor: Pago por Factor:

Mínimo de Siembra: Total Siembra:

Evaluador:

Fecha de Evaluación:

392 x 392

Enviar a Positiva

Imagen 16: Prototipo Enviar Aviso a Positiva
Fuente: Creación propia

2. HISTORIA DE USUARIO N°12: ENVIAR AVISO AL AJUSTADOR

HISTORIA DE USUARIO			
IDENTIFICADOR:	012		
NOMBRE:	Enviar Aviso al Ajustador		
PRIORIDAD:	Alta	USUARIO:	Positiva
ESTIMACIÓN:	5	ITERACIÓN:	04
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Positiva Quiero: Enviar los avisos ya corregidos al Ajustador Para: Que procedan con la verificación y evaluación.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		1	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Enviar Aviso			
Dado: que los avisos ya se encuentran en el estado POSITIVA Cuando: el usuario Positiva quiera enviar el aviso al Ajustador Entonces: El sistema mostrará la opción para que el aviso sea enviado al Ajustador.			
Escenario 2: No se puede enviar aviso			
Dado: que los avisos ya se encuentran en el estado POSITIVA Cuando: el usuario Positiva quiera enviar el aviso al Ajustador y desee enviar Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está enviando el aviso.			

Tabla 13: HU N ° 12: Enviar Aviso al Ajustador
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU12 se muestra en la imagen 17, la cual es el siguiente:

Imagen 17: Prototipo Enviar Avisos al Ajustador
Fuente: Creación propia

3. HISTORIA DE USUARIO N°13: ENVIAR AVISO AL EVALUADOR

HISTORIA DE USUARIO			
IDENTIFICADOR:	013		
NOMBRE:	Enviar Aviso al Evaluador		
PRIORIDAD:	Alta	USUARIO:	Ajustador
ESTIMACIÓN:	5	ITERACIÓN:	04
DESARROLLADORES:	<ul style="list-style-type: none">Cinthya AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Ajustador Quiero: Programar y enviar los avisos al Evaluador Para: Que procedan con la verificación y evaluación.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		1	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Enviar Aviso			
Dado: que los avisos ya se encuentran en el estado Ajustador Cuando: el ajustador programe la fecha de evaluación y asigne evaluador y desee enviar Entonces: El sistema mostrará la opción para que el aviso sea enviado al evaluador.			
Escenario 3: No se puede enviar aviso			
Dado: que los avisos ya se encuentran en el estado Ajustador Cuando: el ajustador programe la fecha de evaluación y asigne evaluador y desee enviar Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está enviando el aviso.			

Tabla 14: HU N°13: Enviar Aviso al Evaluador

Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU13 se muestra en la imagen 18, la cual es el siguiente:

Imagen 18: Prototipo Enviar Aviso al Ajustador

Fuente: Creación propia

v. ITERACIÓN V

1. HISTORIA DE USUARIO N° 14: RETORNAR AVISOS AL AJUSTADOR

HISTORIA DE USUARIO			
IDENTIFICADOR:	014		
NOMBRE:	Retornar Avisos al AJUSTADOR		
PRIORIDAD:	Alta	USUARIO:	EVALUADOR
ESTIMACIÓN:	5	ITERACIÓN:	05
DESARROLLADORES:	<ul style="list-style-type: none">Cinthya AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Quiero: Retornar los avisos mal ingresados al Ajustador. Para: Su corrección.			EVALUADOR
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		1	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Aviso incorrecto			
Dado: que los datos no están conformes Cuando: El evaluador ingrese al aviso Entonces: el sistema mostrará una opción de retorno al ajustador			
Escenario 2: solicito cambio de evaluador			
Dado: que el aviso ya tiene asignado un evaluador Cuando: El Evaluador ingrese al aviso y no pueda atenderlo, debe retornar al ajustador para que vuelva a asignar evaluador Entonces: el sistema mostrará una opción de retorno al ajustador.			
Escenario 3: No se puede retornar el aviso			
Dado: que los avisos ya se encuentran en el estado Evaluador Cuando: el evaluador ingrese al aviso y desee retornar al ajustador Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está retornando el aviso.			

Tabla 15: HU N°14: Retornar Avisos al AJUSTADOR
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU14 se muestra en la imagen 19, la cual es el siguiente:

Imagen 19: Prototipo Retornar Aviso al Ajustador
Fuente: Creación propia

2. HISTORIA DE USUARIO N° 15: RETORNAR AVISOS A POSITIVA

HISTORIA DE USUARIO			
IDENTIFICADOR:	015		
NOMBRE:	Retornar Avisos a Positiva.		
PRIORIDAD:	Alta	USUARIO:	AJUSTADOR
ESTIMACIÓN:	5	ITERACIÓN:	05
DESARROLLADORES:	<ul style="list-style-type: none">Cintha AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo:		AJUSTADOR.	
Quiero: retornar los avisos mal ingresados			
Para: Su corrección.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		1	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Aviso incorrecto			
Dado: que los datos no están conformes			
Cuando: El ajustador ingrese al aviso retornado			
Entonces: el sistema mostrará una opción de retorno al a Positiva.			
Escenario 2: No se puede retornar el aviso			
Dado: que los avisos ya se encuentran en el estado Ajustador			
Cuando: el ajustador ingrese al aviso y desee retornar a Positiva			
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está retornando el aviso.			

Tabla 16: HU N°15: Retornar Avisos a POSITIVA
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU15 se muestra en la imagen 20, la cual es el siguiente:

Imagen 20: Prototipo: Retornar Aviso a POSITIVA
Fuente: Creación propia

3. HISTORIA DE USUARIO N°16: RETORNAR AVISOS A MINAGRI

HISTORIA DE USUARIO			
IDENTIFICADOR:	016		
NOMBRE:	Retornar Avisos a MINAGRI		
PRIORIDAD:	Alta	USUARIO:	POSITIVA
ESTIMACIÓN:	5	ITERACIÓN:	05
DESARROLLADORES:	<ul style="list-style-type: none">Cinthya AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Quiero: retornar los avisos mal ingresados Para: Su corrección.			POSITIVA
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		1	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Aviso incorrecto			
Dado: que los datos no están conformes Cuando: el Positiva ingrese al aviso y desee retornar a MINAGRI Entonces: El sistema mostrará un mensaje de “Se retornó con éxito”			
Escenario 2: No se puede retornar el aviso			
Dado: que los avisos ya se encuentran en el estado Positiva Cuando: el Positiva ingrese al aviso y desee retornar a MINAGRI Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está retornando el aviso.			

Tabla 17: HU N°16: Retornar Avisos a MINAGRI
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU16 se muestra en la imagen 21, la cual es el siguiente:

Imagen 21: Prototipo Regresar Avisos al MINAGRI
Fuente: Creación propia

4. HISTORIA DE USUARIO N°17: RETORNAR AVISOS AL REPRESENTANTE DE REGIÓN

HISTORIA DE USUARIO			
IDENTIFICADOR:	017		
NOMBRE:	Retornar Avisos al REPRESENTANTE DE REGIÓN		
PRIORIDAD:	Alta	USUARIO:	MINAGRI
ESTIMACIÓN:	5	ITERACIÓN:	05
DESARROLLADORES:	<ul style="list-style-type: none">• Cinthya Amaya• Christian Rodas		
DESCRIPCIÓN			
Cómo: Quiero: retornar los avisos mal ingresados Para: Su corrección.			MINAGRI
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		1	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Aviso incorrecto			

Dado: que los datos no están conformes
Cuando: MINAGRI de clic en el botón RETORNAR A REPRESENTANTE DE REGIÓN
Entonces: el sistema enviará el aviso a REPRESENTANTE DE REGIÓN para su revisión.
Escenario 2: No se puede retornar el aviso
Dado: que los avisos ya se encuentran en el estado MINAGRI
Cuando: el MINAGRI ingrese al aviso y desee retornar a Representante de Región
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está retornando el aviso.

Tabla 18: HU N°17: Retornar Avisos al REPRESENTANTE DE REGIÓN
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU17 se muestra en la imagen 22, la cual es el siguiente:

Detalle del aviso

Detalle Puntos

Departamento: Provincia:

Distrito: Sector Estadístico:

Cultivo: Tipo de Evento:

Estado: **MINAGRI**

Factor: Pago por Factor:

Mínimo de Siembra: Total Siembra:

Evaluador:

Fecha de Evaluación:

392 x 392

Retornar a Jefe de Región **Enviar a Positiva**

Imagen 22: Prototipo: Retornar Avisos al Representante de Región
Fuente: Creación propia

vi. ITERACIÓN VI

1. HISTORIA DE USUARIO N°18: ASIGNAR EVALUADOR

HISTORIA DE USUARIO			
IDENTIFICADOR:	018		
NOMBRE:	Asignar Evaluador		
PRIORIDAD:	Alta	USUARIO:	Ajustador
ESTIMACIÓN:	6	ITERACIÓN:	06
DESARROLLADORES:	<ul style="list-style-type: none">Cinthya AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo:			Ajustador
Quiero: Asignar Evaluador.			
Para: que cada aviso ya tenga asignado un evaluador.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		2	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Asignar Evaluador			
Dado: que se tiene que asignar un evaluador a cada aviso			
Cuando: el ajustador ingrese al aviso y asigne evaluador			
Entonces: el sistema mostrará un mensaje “Se asignó con éxito”			
Escenario 2: No encuentro el Evaluador que deseo asignar			
Dado: que se tiene que asignar un evaluador a cada aviso			
Cuando: el ajustador ingrese al aviso y no encuentra en el combo el evaluador que solicita			
Entonces: el sistema solo mostrará los evaluadores que brindó Agroevaluaciones.			
Escenario 3: No se puede guardar el evaluador Asignado			
Dado: que se tiene que asignar un evaluador a cada aviso			
Cuando: el ajustador ingrese al aviso y asigne evaluador			
Entonces: el sistema mostrará un mensaje “No se puede asignar evaluador”			

Tabla 19: HU N°18: Asignar Evaluador
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU18 se muestra en la imagen 23, la cual es el siguiente:

Imagen 23: Prototipo Asignar Evaluador
Fuente: Creación propia

2. HISTORIA DE USUARIO N°19: PROGRAMAR EVALUACIÓN

HISTORIA DE USUARIO			
IDENTIFICADOR:	019		
NOMBRE:	Programar Evaluación		
PRIORIDAD:	Alta	USUARIO:	Ajustador
ESTIMACIÓN:	6	ITERACIÓN:	06
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo:			Ajustador
Quiero: programar fecha de evaluación			
Para: que se proceda con la evaluación del aviso.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		2	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Guardar fecha de Programación de aviso			
Dado: que el sistema solicita una fecha de programación de evaluación			
Cuando: el ajustador ingrese una fecha mayor a la actual			
Entonces: El sistema mostrará un mensaje “Se Programó con éxito”			
Escenario 2: Fecha de programación menor a la fecha actual			

Dado: que el sistema solicita una fecha de programación de evaluación
Cuando: el ajustador ingrese una fecha menor a la actual
Entonces: El sistema mostrará un mensaje de error “Fecha no válida”
Escenario 2: No se puede Guardar Fecha de programación
Dado: que los avisos necesitan información importante por parte de POSITIVA
Cuando: POSITIVA ingrese a los avisos podrá ingresar los datos necesarios.
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está guardando los datos.

Tabla 20: HU N°19: Programar Evaluación
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU19 se muestra en la imagen 24, la cual es el siguiente:

Imagen 24: Prototipo Programar Evaluación
Fuente: Creación propia

3. HISTORIA DE USUARIO N°20: AGREGAR DATOS A AVISO

HISTORIA DE USUARIO			
IDENTIFICADOR:	020		
NOMBRE:	Agregar Datos a Aviso		
PRIORIDAD:	Alta	USUARIO:	Positiva
ESTIMACIÓN:	8	ITERACIÓN:	06
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo:			Positiva
Quiero: Agregar datos al aviso.			
Para: que al cerrar el aviso pueda generar el dictamen.			
OBSERVACIONES			
Ninguna.			

TAREAS	
Descripción	Duración (horas)
Analizar, diseñar y validar interfaz de usuario	1
Diseñar estructura para la base de datos	1
Crear estructuras para base de datos	1
Desarrollar, construir lógica de negocio e interfaz de usuario	4
Realizar pruebas y corregir errores	1
CRITERIOS DE ACEPTACIÓN	
Escenario 1: Guardar factor, pago de factor y Mínimo siembra	
Dado: que los avisos necesitan información importante por parte de POSITIVA	
Cuando: POSITIVA haga clic en las cajas de texto factor, pago de factor y mínimo siembra podrá ingresar los datos necesarios.	
Entonces: El sistema guardará esos datos que luego serán enviados al ajustador.	
Escenario 2: No se puede Guardar factor, pago de factor y Mínimo siembra	
Dado: que los avisos necesitan información importante por parte de POSITIVA	
Cuando: POSITIVA ingrese a los avisos podrá ingresar los datos necesarios.	
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está guardando los datos.	

Tabla 21: HU N°20: Agregar datos a aviso
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU20 se muestra en la imagen 25, la cual es el siguiente:

Detalle del aviso

Detalle Puntos

Departamento: Provincia:

Distrito: Sector Estadístico:

Cultivo: Tipo de Evento:

Estado: **Positiva**

Factor: Pago por Factor:

Mínimo de Siembra: Total Siembra:

Evaluador:

Fecha de Evaluación:

392 x 392

Imagen 25: Protocolo Agregar Datos a Aviso
Fuente: Creación propia

4. HISTORIA DE USUARIO N°21: EVALUAR AVISOS VÍA WEB

HISTORIA DE USUARIO			
IDENTIFICADOR:	021		
NOMBRE:	Evaluar avisos vía web		
PRIORIDAD:	Alta	USUARIO:	Evaluador
ESTIMACIÓN:	8	ITERACIÓN:	06
DESARROLLADORES:	<ul style="list-style-type: none">Cinthya AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Quiero: Evaluar los avisos vía web Para: Tener datos de evaluación del aviso.			Evaluador
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		4	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Guardar Datos de evaluación			
Dado: que puede pasar algún percance con la Tablet.			
Cuando: el evaluador ingresará al aviso y procederá a evaluar			
Entonces: el sistema mostrará un mensaje indicando: “Se guardaron los datos con éxito”			
Escenario 2: No se puede Guardar Datos de evaluación			
Dado: que puede pasar algún percance con la Tablet.			
Cuando: el evaluador ingresará al aviso y procederá a evaluar			
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se está guardando los datos.			
Escenario 3: El aviso no puede ser evaluado			
Dado: que el aviso no se podía evaluar por condiciones del cultivo			
Cuando: evaluador ingrese al aviso podrá desestimar			
Entonces: El sistema mostrará un mensaje indicando “El aviso fue desestimado”			

Tabla 22: HU N° 22: Evaluar avisos vía web
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU21 se muestra en la imagen 26, la cual es el siguiente:

Detalle de Aviso

Detalle

Puntos

[Foto]	Item	Posición Este	Posición Norte	Registra Análisis	N° Plantas < 100	N° Cultivos por Planta <25	N° Cultivos Por Análisis	Siembra
[Foto]	1			<input checked="" type="checkbox"/>				
[Foto]	2			<input type="checkbox"/>				
[Foto]	3			<input checked="" type="checkbox"/>				
[Foto]	4			<input type="checkbox"/>				

TOTAL :

Retornar Ajustador

Guardar

Desestimar

Cerrar Aviso

Imagen 26:Prototipo Evaluar avisos vía web

Fuente: Creación propia

vii. ITERACIÓN VII

HISTORIA DE USUARIO			
IDENTIFICADOR:	022		
NOMBRE:	Inicio de Sesión		
PRIORIDAD:	Alta	USUARIO:	
ESTIMACIÓN:	6	ITERACIÓN:	07
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Positiva / Minagri/ Ajustador / Evaluador/ Representante de Región/ Evaluadores Quiero: Ingresar al Sistema de Gestión, Análisis y Cartografía para el Seguro Agrícola Catastrófico Para: Registrar, retornar y agregar datos al aviso.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		2	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Ingresar al sistema			
Dado: que se solicita ingresar al sistema			
Cuando: Los usuarios ingresen al sistema web y digiten su usuario y clave			
Entonces: el sistema permitirá el ingreso al usuario.			
Escenario 2: No se puede iniciar sesión			
Dado: que se necesita registrar y hacer seguimiento respectivo a los avisos			
Cuando: Los usuarios ingresen al sistema web y digiten su usuario y clave			
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se puede iniciar sesión.			

1. HISTORIA DE USUARIO N°22: INICIO DE SESIÓN

Tabla 23: HU N° 22: Inicio de Sesión

Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU22 se muestra en la imagen 27, la cual es el siguiente:

Imagen 27: Prototipo Inicio de Sesión

Fuente: Creación propia

2. HISTORIA DE USUARIO N°23: BUSCAR AVISO

HISTORIA DE USUARIO			
IDENTIFICADOR:	023		
NOMBRE:	Buscar Aviso		
PRIORIDAD:	Media	USUARIO:	
ESTIMACIÓN:	7	ITERACIÓN:	07
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Positiva / Minagri/ Ajustador / Evaluador/ Representante de Región			
Quiero: Encontrar avisos buscando solo algún parámetro.			
Para: ubicar el aviso de manera inmediata.			
OBSERVACIONES			
Podrán realizar la búsqueda según permisos.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		3	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Ubicar un aviso			
Dado: que se solicita un aviso puntual			
Cuando: el usuario ingrese la palabra clave por el cual quiere ubicar el aviso y luego de clic en el botón buscar.			
Entonces: el sistema mostrará todos los avisos con esas características.			

Tabla 24: HU N°23: Buscar Aviso
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU23 se muestra en la imagen 28, la cual es el siguiente:

Listar Aviso

Nombre del Usuario:

USUARIO

Buscar....

Buscar

Cultivo	Región	Distrito	Provincia	Sector Estadístico	Evento	Creado	Resultado	Estado	Detalle
							En Proceso	MINAGRI	VER DETALLE
							En Proceso	Positiva	VER DETALLE
							En Proceso	Ajustador	VER DETALLE
							En Proceso	Evaluador	VER DETALLE
							Indemnizable	Finalizó	VER DETALLE
							No Indemnizable	Finalizó	VER DETALLE

Imagen 28: Prototipo: Buscar Aviso
Fuente: Creación propia

3. HISTORIA DE USUARIO N°24: EXPORTAR AVISOS EN EXCEL

HISTORIA DE USUARIO			
IDENTIFICADOR:	024		
NOMBRE:	Exportar Avisos en Excel		
PRIORIDAD:	Alta	USUARIO:	
ESTIMACIÓN:	8	ITERACIÓN:	07
DESARROLLADORES:	<ul style="list-style-type: none">Cinthy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Positiva / Minagri/ Ajustador / Representante de Región			
Quiero: Exportar en formato Excel todos los avisos o solo los que yo indique en la búsqueda.			
Para: tener un mejor control			
OBSERVACIONES			
Sólo Positiva, MINAGRI y Ajustador podrán exportar todos los avisos de todas las regiones, Los Representantes de Región sólo podrán descargar los avisos de su jurisdicción, el evaluador solo podrá exportar en Excel los avisos que le han sido asignados.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		4	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: exportar avisos en Excel			
Dado: que los avisos se necesitan en Excel			
Cuando: el usuario ingrese a los avisos que necesita exportar			
Entonces: El sistema exportará lo que se haya seleccionado.			
Escenario 2: No puedo exportar los avisos en Excel			
Dado: que los avisos se necesitan en Excel			
Cuando: el usuario ingrese a los avisos que necesita exportar			
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se puede descargar el archivo.			
Escenario 3: No se seleccionó ningún aviso			
Dado: no se indicó que avisos se desean exportar			
Cuando: el usuario cuando solicite exportar			
Entonces: El sistema mostrará un error “No Hay avisos seleccionados para exportar”			

Tabla 25: HU N°24: Exportar avisos en Excel
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU24 se muestra en la imagen 29, la cual es el siguiente:

Listar Aviso

Nombre del Usuario:

USUARIO

Buscar...

Buscar



Cultivo	Región	Distrito	Provincia	Sector Estadístico	Evento	Creado	Resultado	Estado	Detalle
							En Proceso	MINAGRI	VER DETALLE
							En Proceso	Positiva	VER DETALLE
							En Proceso	Ajustador	VER DETALLE
							En Proceso	Evaluador	VER DETALLE
							Indemnizable	Finalizó	VER DETALLE
							No Indemnizable	Finalizó	VER DETALLE

Imagen 29: Prototipo Exportar a Excel
Fuente: Creación propia

viii. ITERACIÓN VIII

1. HISTORIA DE USUARIO N° 25: DESCARGAR ACTAS DE ATENCIÓN POR AVISO

HISTORIA DE USUARIO			
IDENTIFICADOR:	025		
NOMBRE:	Descargar Actas de atención por Aviso		
PRIORIDAD:	Alta	USUARIO:	
ESTIMACIÓN:	8	ITERACIÓN:	08
DESARROLLADORES:	<ul style="list-style-type: none">Cinthya AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Positiva / Minagri/ Ajustador / Evaluador Quiero: exportar el acta de atención de los avisos cerrados. Para: tener un mejor control.			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		4	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Generar Acta en PDF			

Dado: que se debe contar con un acta por aviso
Cuando: el usuario ingrese al aviso cerrado y descargue el aviso
Entonces: el sistema mostrará un mensaje que indica “Se descargó con éxito”
Escenario 2: No puedo Generar Acta en PDF
Dado: que se debe contar con un acta por aviso
Cuando: el usuario ingrese al aviso cerrado y descargue el aviso
Entonces: El sistema mostrará un mensaje de error indicando el inconveniente por el cual no se puede descargar el archivo

Tabla 26: HU N° 25: Descargar actas de atención por aviso
Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU25 se muestra en la imagen 30, la cual es el siguiente:

Imagen 30: Prototipo Descargar Actas de atención por aviso
Fuente: Creación propia

2. HISTORIA DE USUARIO N° 26: VER ESTADO DE AVISOS

HISTORIA DE USUARIO			
IDENTIFICADOR:	026		
NOMBRE:	Ver estado de avisos.		
PRIORIDAD:	Alta	USUARIO:	
ESTIMACIÓN:	8	ITERACIÓN:	08
DESARROLLADORES:	<ul style="list-style-type: none">Cinhy AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Positiva / Minagri/ Ajustador / Representante de Región			
Quiero: Ver el estado de los avisos, en que usuario se encuentra.			
Para: tener un mejor control			
OBSERVACIONES			
Ninguna.			

TAREAS	
Descripción	Duración (horas)
Analizar, diseñar y validar interfaz de usuario	1
Diseñar estructura para la base de datos	1
Crear estructuras para base de datos	1
Desarrollar, construir lógica de negocio e interfaz de usuario	4
Realizar pruebas y corregir errores	1
CRITERIOS DE ACEPTACIÓN	
Escenario 1: Estado de los avisos	
Dado: que se solicita saber el estado de los avisos	
Cuando: el usuario ingrese al sistema a revisar los estados de los avisos	
Entonces: El sistema mostrará dashboard de estados y mostrará los datos según la selección del usuario.	

Tabla 27: HU N°26: Ver estado de aviso

Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU26 se muestra en la imagen 31, la cual es el siguiente:

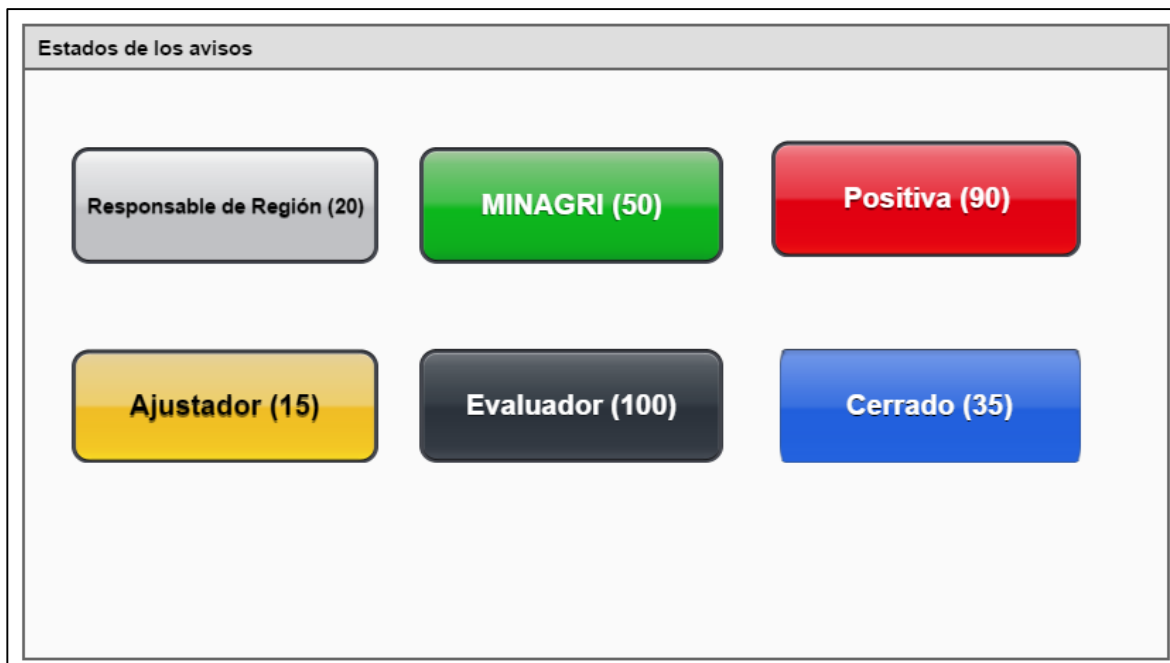


Imagen 31: Prototipo Ver estado de los avisos

Fuente: Creación propia

3. HISTORIA DE USUARIO N° 27: LISTAR AVISOS


HISTORIA DE USUARIO			
IDENTIFICADOR:	027		
NOMBRE:	Listar avisos		
PRIORIDAD:	Media	USUARIO:	
ESTIMACIÓN:	8	ITERACIÓN:	08
DESARROLLADORES:	<ul style="list-style-type: none">Cinthya AmayaChristian Rodas		
DESCRIPCIÓN			
Cómo: Positiva / Minagri/ Ajustador / Representante de Región			
Quiero: Ver todos los avisos que han sido ingresados al sistema.			
Para: tener un mejor control			
OBSERVACIONES			
Podrán ver los avisos según permisos.			
TAREAS			
Descripción		Duración (horas)	
Analizar, diseñar y validar interfaz de usuario		1	
Diseñar estructura para la base de datos		1	
Crear estructuras para base de datos		1	
Desarrollar, construir lógica de negocio e interfaz de usuario		4	
Realizar pruebas y corregir errores		1	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Ver todos los avisos			
Dado: que se solicita listar todos los avisos			
Cuando: el usuario de clic al ítem listar avisos			
Entonces: el sistema mostrará todos los avisos.			

Tabla 28: HU N° 27: Listar avisos

Fuente: creación Propia a través de la información brindada por el cliente

El prototipo de la HU27 se muestra en la imagen 32, la cual es el siguiente:

Listar Aviso



PROCESOS:

Listar Avisos:

Nombre del Usuario:				USUARIO					
Cultivo	Región	Distrito	Provincia	Sector Estadístico	Evento	Creado	Resultado	Estado	Detalle
							En Proceso	MINAGRI	VER DETALLE
							En Proceso	Positiva	VER DETALLE
							En Proceso	Ajustador	VER DETALLE
							En Proceso	Evaluador	VER DETALLE
							Indemnizable	Finalizó	VER DETALLE
							No Indemnizable	Finalizó	VER DETALLE

Imagen 32: Prototipo Listar Aviso

Fuente: Creación propia

B. REQUERIMIENTOS NO FUNCIONALES

En la tabla 29 indicaremos todos los requisitos no funcionales que debemos considerar para el funcionamiento eficiente y eficaz de nuestro sistema:

ID	REQUERIMIENTO	DESCRIPCIÓN
ReqNoFun01	Ingreso de datos de forma manual de factor, pago por factor y mínimo de siembra	Ingreso de datos por parte de Positiva
ReqNoFun02	Fácil interacción de los usuarios con el sistema web y el aplicativo móvil	Los sistemas tanto el web como el App Android debe ser fácil de usar, con ayuda de interfaces intuitivas.
ReqNoFun03	Estabilidad del sistema cuando se procese grandes cantidades de información.	El sistema deberá dar soporte a gran cantidad de información durante su proceso
ReqNoFun04	Seguridad en autenticación de usuario y contraseña en el ingreso del sistema	Se debe establecer un nivel de seguridad, en el que el sistema permita a los usuarios el ingreso hacia las diversas funcionalidades, accediendo solo a datos de acuerdo con el rol o perfil del usuario.
ReqNoFun05	Corroborar los atributos de acuerdo con el tipo y datos obligatorios	El sistema debe validar el ingreso de los datos de los usuarios de acuerdo con el tipo al cual pertenece el atributo e identificar si los datos obligatorios están ingresados.
ReqNoFun06	Ejecución de reglas de negocio	Se debe cumplir el flujo del sistema sin ninguna excepción.
ReqNoFun7	Manejo de Contraseñas	El sistema debe brindar las siguientes funcionalidades de autenticación:
		Mostrar interfaz para autenticación de usuarios en el sistema (login y password)
		Ningún usuario puede cambiar su contraseña, sólo el super administrador
		Identificar el error que comete el usuario si se producen el proceso de autenticación. Y mostrar mediante un mensaje de error.

Tabla 29: Requerimientos no funcionales del sistema
Fuente: creación Propia a través de la información brindada por el cliente

II. DISEÑO DEL PROYECTO

a. DISEÑO LÓGICO

En la imagen 33 que está a continuación, indicamos cual es el diseño lógico de nuestro sistema:

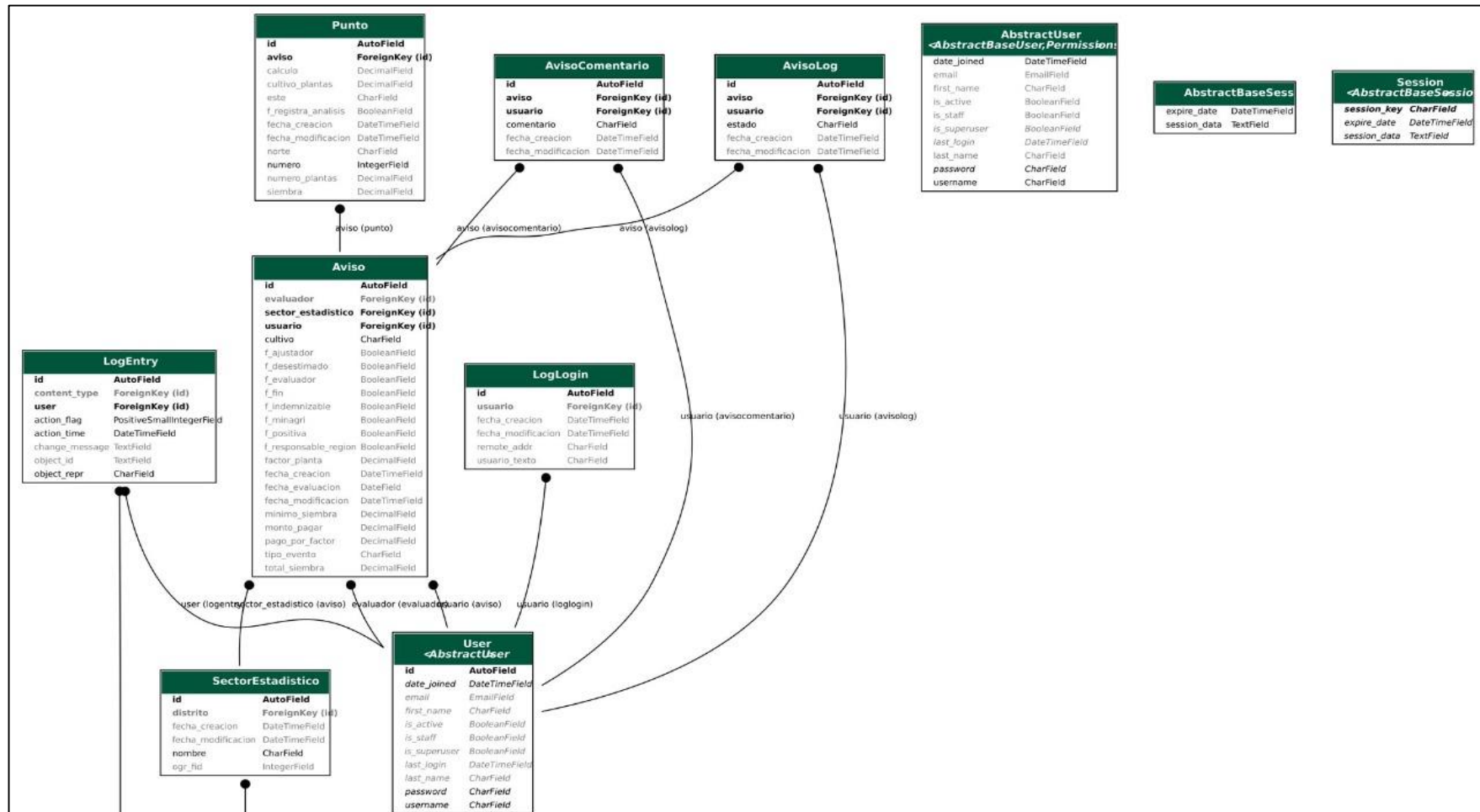


Imagen 33: Diagrama Lógico 1 era Parte

Fuente: creación Propia a través de la información brindada por el cliente

En la imagen 34 mostramos la continuación del diseño lógico de nuestro sistema:

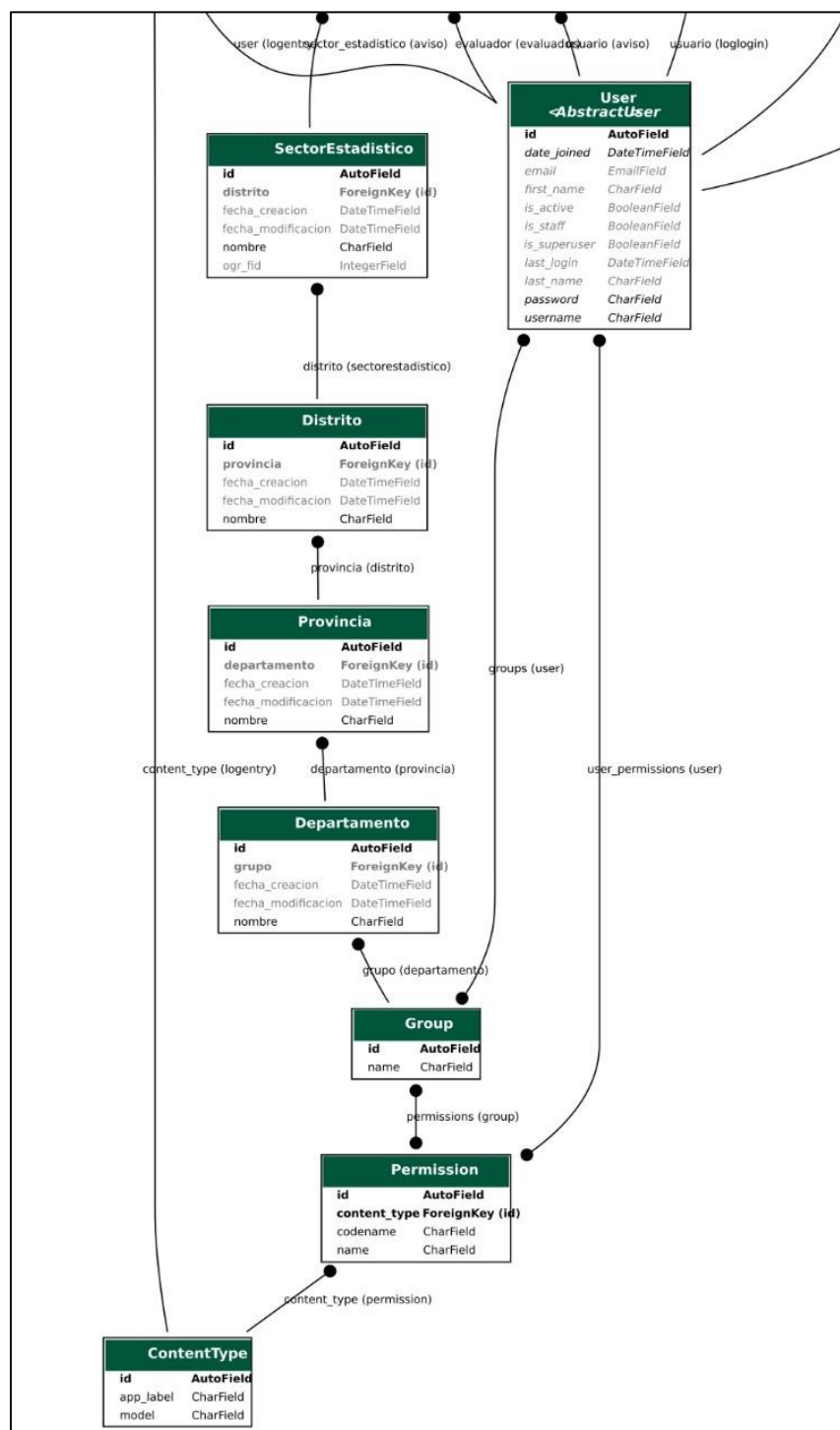


Imagen 34: Diagrama Lógico 2 da Parte

Fuente: creación Propia a través de la información brindada por el cliente

b. MODELO DEL NEGOCIO

En la imagen 35 que está a continuación, indicamos cual es modelo del negocio mediante sus procesos:

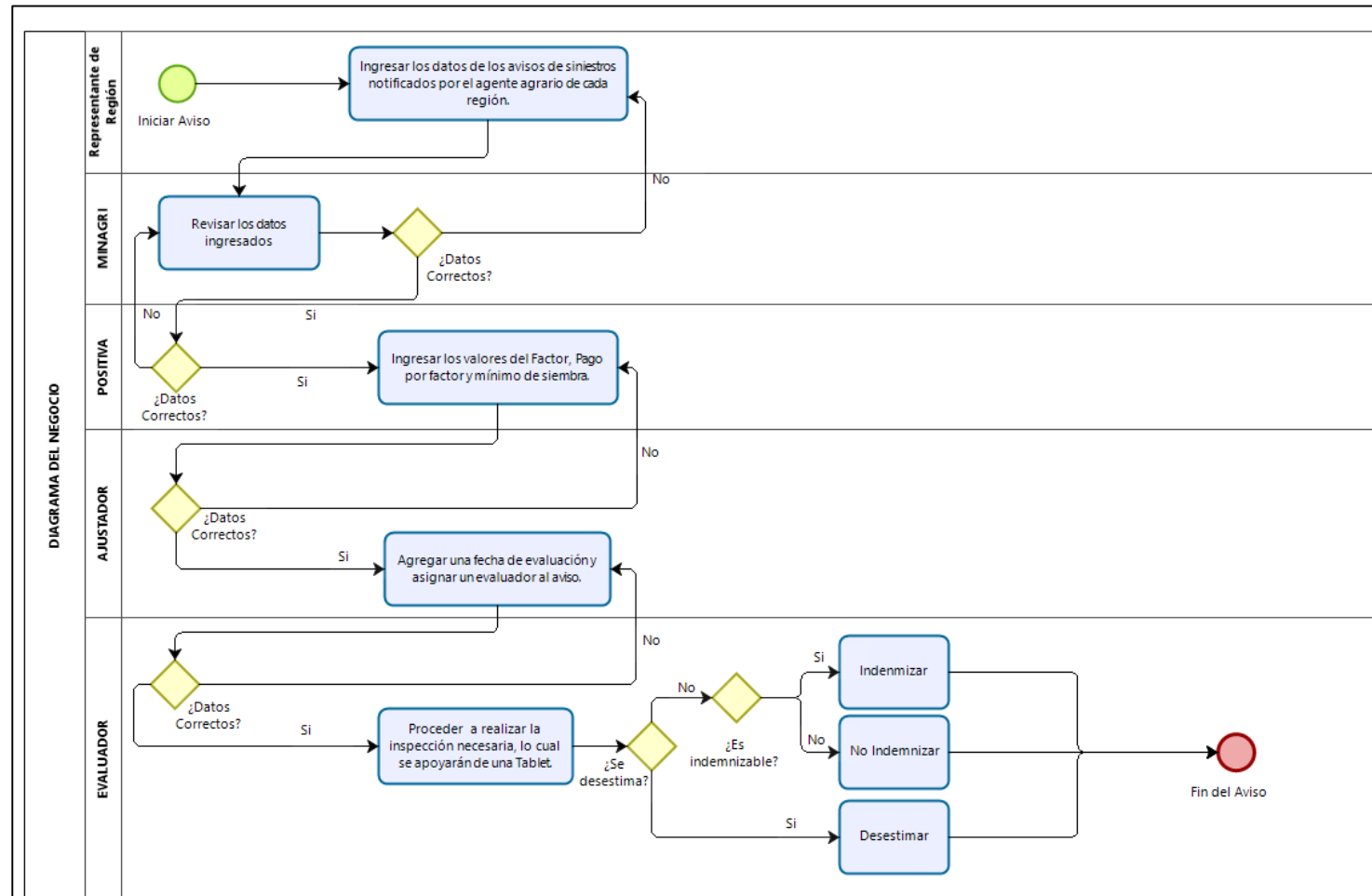


Imagen 35 : Modelo de Negocio
Fuente: creación Propia a través de la información brindada por el cliente

c. DIAGRAMA DE ESTADOS

En la imagen 36 que está a continuación, indicamos cual es diagrama de estados de nuestro sistema:

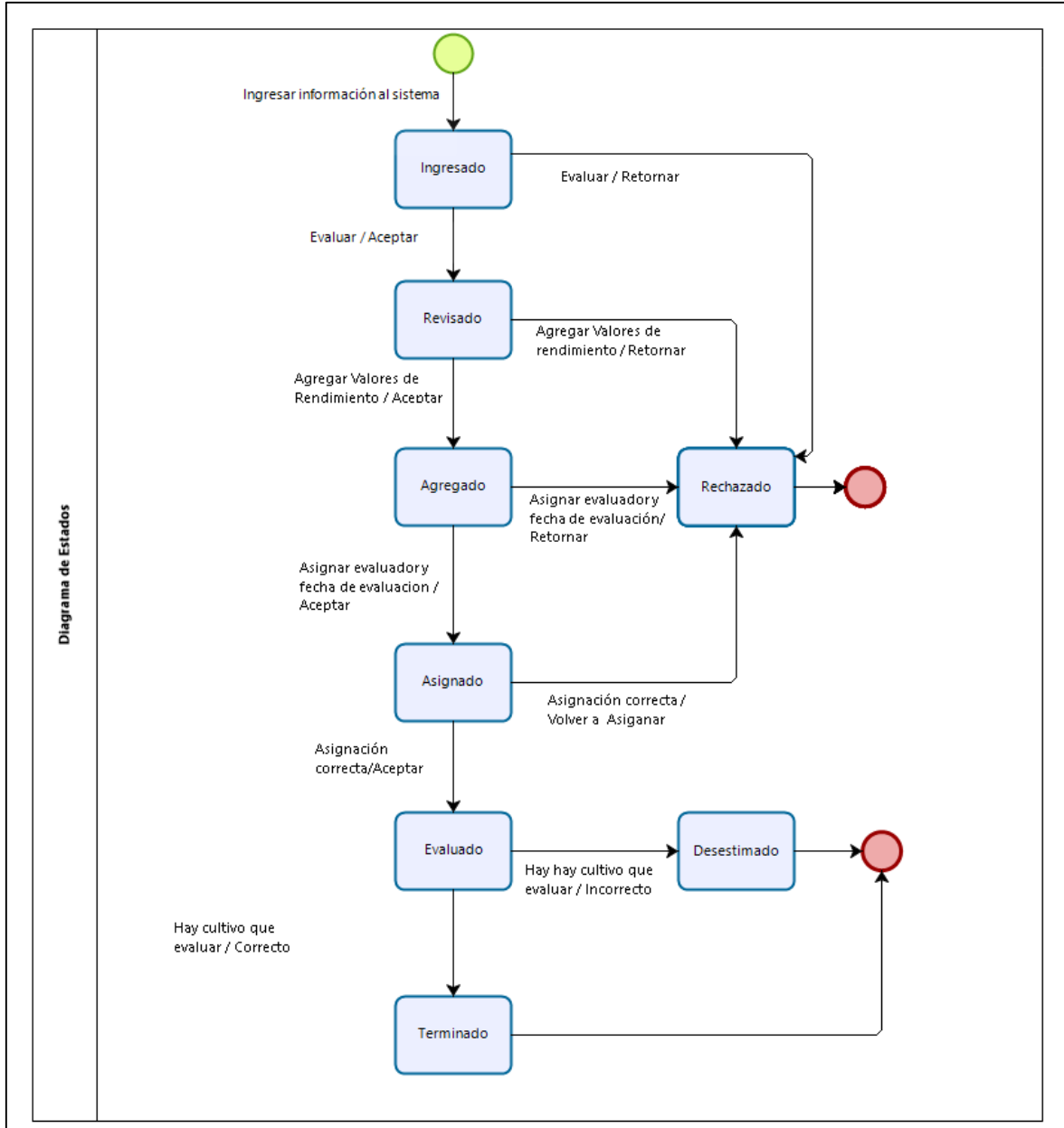


Imagen 36: Diagrama de Estados
Fuente: creación Propia a través de la información brindada por el cliente

d. MODELO DE BASE DE DATOS

En la imagen 37 que está a continuación, indicamos cual es el modelo de base de datos de datos de nuestro sistema:

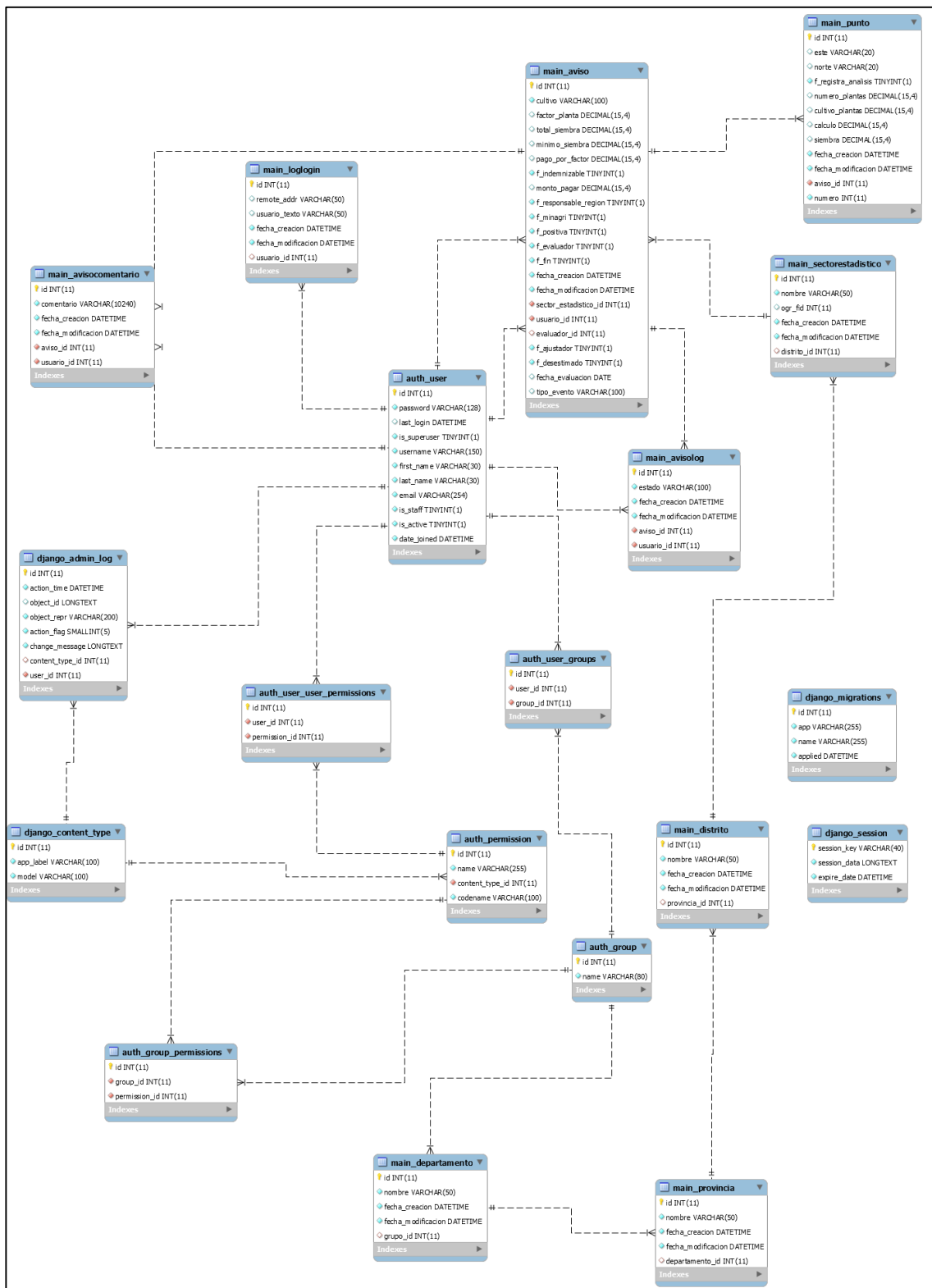


Imagen 37: Modelo de base de datos

Fuente: creación Propia a través de la información brindada por el cliente

e. DIAGRAMA DE COMPONENTES

En la imagen 38 que está a continuación, indicamos cual es el diagrama de componentes de nuestro sistema:

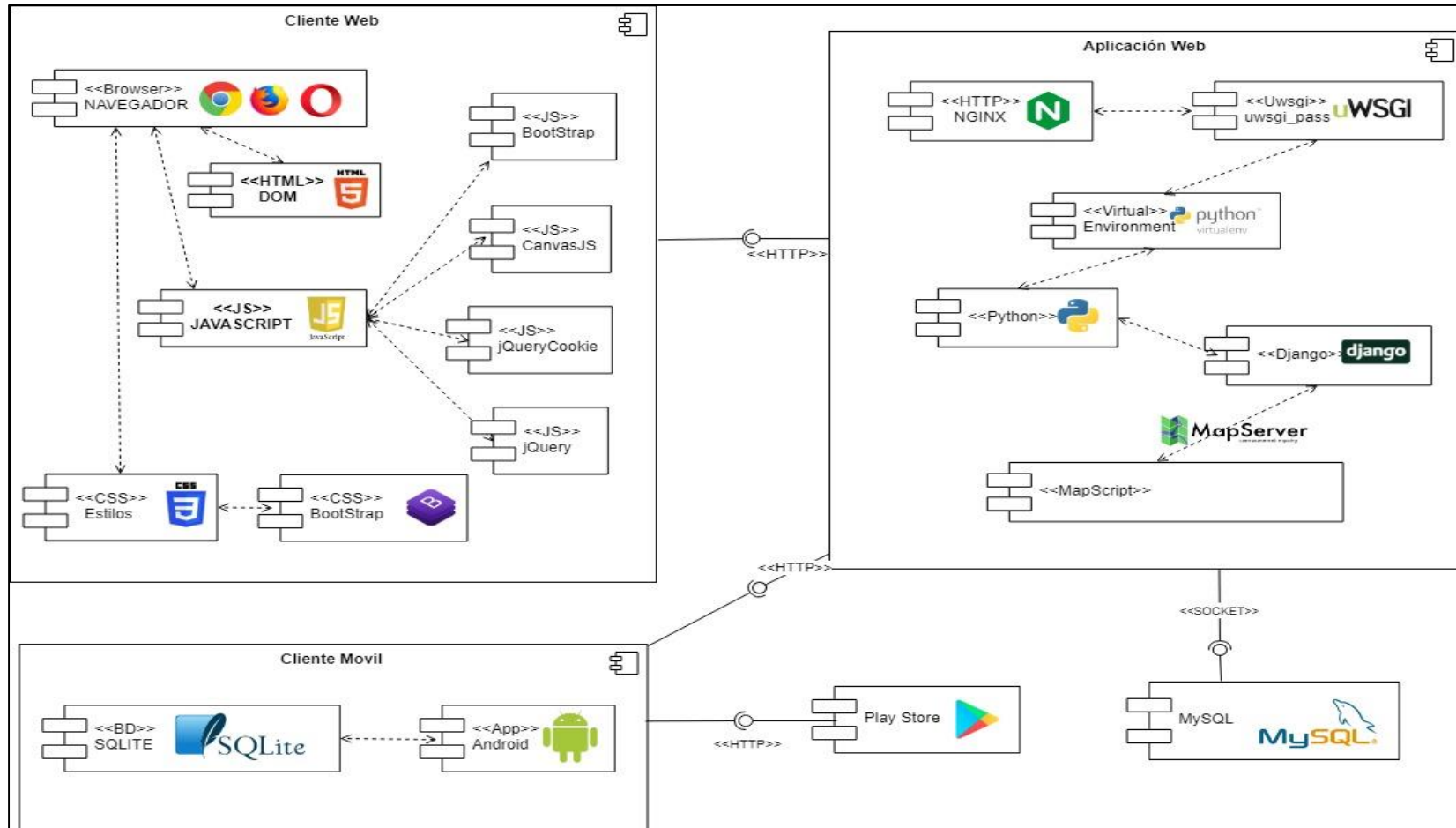


Imagen 38: Diagrama de Componentes
Fuente: creación Propia a través de la información brindada por el cliente

f. DIAGRAMA DE DESPLIEGUE

En la imagen 39 que está a continuación, indicamos cual es el diagrama de despliegue de nuestro sistema:

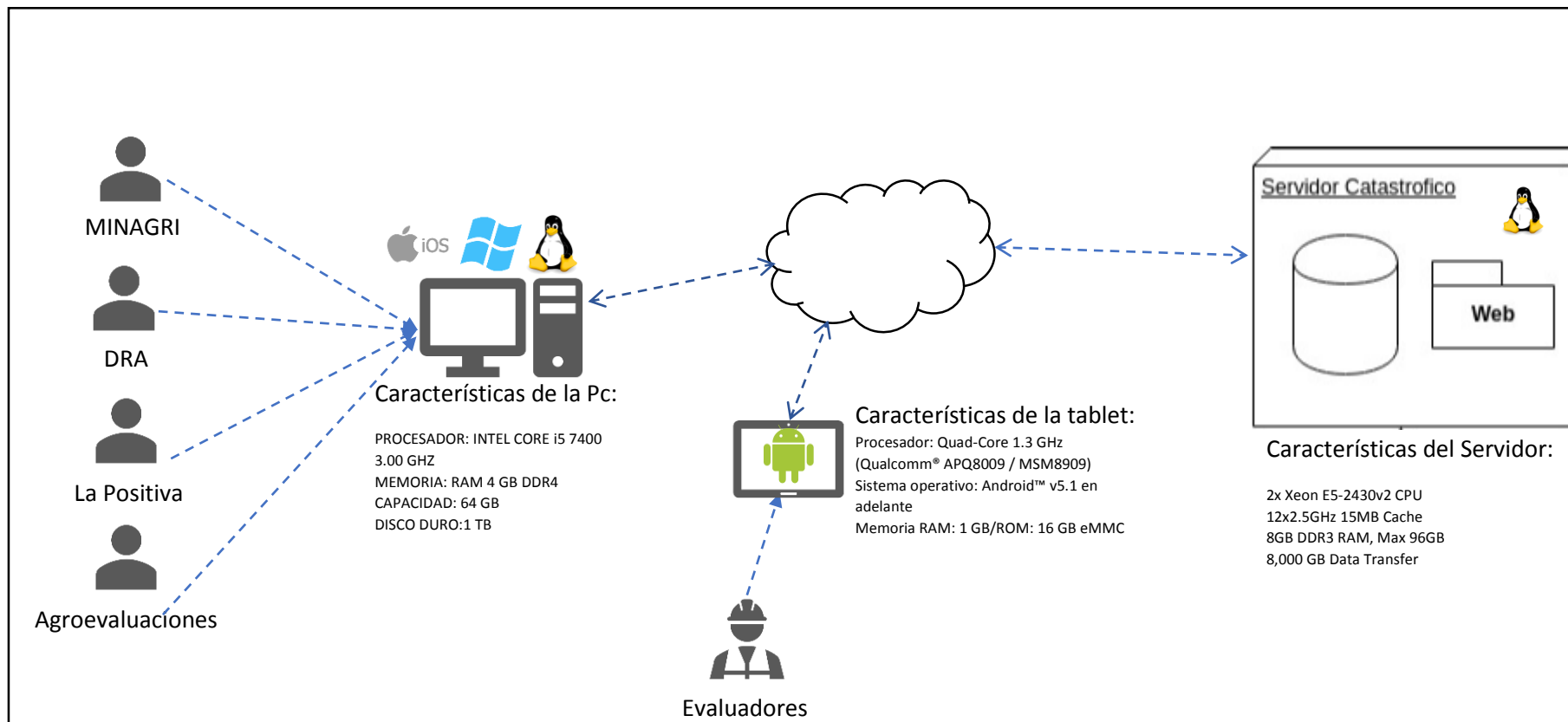


Imagen 39: Diagrama de Despliegue

Fuente: creación Propia a través de la información brindada por el cliente

A. DICCIONARIO DE DATOS

i. GRUPOS O ROLES DEL APLICATIVO

La tabla 30 se crea automáticamente con el framework Django bajo el nombre de `auth_group` y la cual nos permite definir los grupos o roles del aplicativo.

Tabla: <code>auth_group</code>						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
<code>id</code>	<code>int (11)</code>	<code>NULL</code>	<code>PRI</code>	<code>NO</code>	<code>auto_increment</code>	Identificador de tuplas de la tabla.
<code>name</code>	<code>varchar (80)</code>	<code>NULL</code>	<code>UNI</code>	<code>NO</code>		Nombre del registro de la tabla.

Tabla 30: Roles del Aplicativo
Fuente: Creación automática DJANGO

ii. PERMISOS DE ACCIONES DE LOS GRUPOS EN EL APLICATIVO

La tabla 31 se crea automáticamente con el framework Django bajo el nombre de `auth_group_permissions`, la cual asigna los permisos a las acciones que realizan los grupos del aplicativo.

Tabla: <code>auth_group_permissions</code>						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
<code>group_id</code>	<code>int (11)</code>	<code>NULL</code>		<code>NO</code>		Referencia a la tabla Grupo, indica que grupo tiene el permiso.
<code>id</code>	<code>int (11)</code>	<code>NULL</code>	<code>PRI</code>	<code>NO</code>	<code>auto_increment</code>	Identificador de tuplas de la tabla.
<code>permission_id</code>	<code>int (11)</code>	<code>NULL</code>		<code>NO</code>		Referencia a la tabla Permiso, indica que permiso tiene el grupo.

Tabla 31: Permisos de los grupos en el aplicativo
Fuente: Creación automática DJANGO

iii. TODAS LAS ACCIONES PERMITIDAS EN EL SISTEMA

La tabla 32 se crea automáticamente con el framework Django bajo el nombre de `auth_permissions`, la cual indica todas las acciones que se pueden realizar en los modelos (tablas) del aplicativo.

Tabla : auth_permission						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
codename	varchar (100)	NULL		NO		Nombre de la acción que se puede realizar sobre el modelo.
content_type_id	int (11)	NULL		NO		Referencia la tabla Content Type.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
name	varchar (255)	NULL		NO		Nombre del registro de la tabla.

Tabla 32: Acciones permitidas en el Sistema
Fuente: Creación automática DJANGO

iv. USUARIOS DE LOS APLICATIVOS

La tabla 33 se crea automáticamente con el framework Django bajo el nombre de auth_user, la cual contiene todos los usuarios activos e inactivos del sistema.

Tabla: auth_user						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
date_joined	datetime	NULL		NO		Una fecha y hora que designa cuándo se creó la cuenta.
email	varchar (254)	NULL		NO		Dirección de correo electrónico.
first_name	varchar (30)	NULL		NO		Nombres
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
is_active	tinyint (1)	NULL		NO		Designa si esta cuenta de usuario debe considerarse activa
is_staff	tinyint (1)	NULL		NO		Designa si este usuario puede acceder al sitio de administración.
is_superuser	tinyint (1)	NULL		NO		Designa que este usuario tiene todos los permisos sin asignarlos explícitamente.
last_login	datetime	NULL		YES		Una fecha y hora del último inicio de sesión del usuario.
last_name	varchar (30)	NULL		NO		Apellidos
password	varchar (128)	NULL		NO		Un hash y metadatos sobre la contraseña.
username	varchar (150)	NULL	UNI	NO		Nombre de usuario para inicio de sesión

Tabla 33: Usuarios de los aplicativos
Fuente: Creación automática DJANGO

v. RELACIÓN ENTRE LOS GRUPOS Y LOS USUARIOS

La tabla 34 se crea automáticamente con el framework Django bajo el nombre de `auth_user_groups`, la cual contiene la relación entre los grupos y los usuarios, es decir qué usuarios están en cada grupo.

Tabla: <code>auth_user_groups</code>						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
<code>group_id</code>	int (11)	NULL		NO		Referencia a la tabla Grupo, indica que grupo tiene el usuario.
<code>id</code>	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
<code>user_id</code>	int (11)	NULL		NO		Referencia a la tabla Usuario, indica que usuario tiene el grupo.

Tabla 34: Relación entre grupos y usuarios
Fuente: Creación automática DJANGO

vi. PERMISOS DE ACCIONES DE LOS USUARIOS EN EL APLICATIVO

La tabla 35 se crea automáticamente con el framework Django bajo el nombre de `auth_user_permissions`, la cual contiene los permisos de acciones de cada usuario en el aplicativo.

Tabla: <code>auth_user_permissions</code>						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
<code>id</code>	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
<code>permission_id</code>	int (11)	NULL		NO		Referencia a la tabla de Permisos, indica que permiso tiene el usuario.
<code>user_id</code>	int (11)	NULL		NO		Referencia a la tabla Usuario, indica que usuario tiene el permiso.

Tabla 35: Permisos a las acciones de los usuarios
Fuente: Creación automática DJANGO

vii. LOG DE LAS ACCIONES QUE SE REALIZAN POR EL ADMINISTRADOR DJANGO (INSERTAR, MODIFICAR, ELIMINAR)

La tabla 36 se crea automáticamente con el framework Django bajo el nombre de `django_admin_log`, la cual contiene el log de las acciones que se realizan por el administrador Django (Insertar, Modificar, Eliminar).

Tabla: django_admin_log						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
action_flag	smallint (5) unsigned	NULL		NO		El tipo de acción registra: ADDITION, CHANGE, DELETION.
action_time	datetime	NULL		NO		La fecha y hora de la acción.
change_message	longtext	NULL		NO		La descripción detallada de la modificación.
content_type_id	int (11)	NULL		YES		El ContentType del objeto modificado.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
object_id	longtext	NULL		YES		La representación textual de la clave primaria del objeto modificado.
object_repr	varchar (200)	NULL		NO		Presentación de cadena "oficial" de un objeto
user_id	int (11)	NULL		NO		Referencia del usuario que realizó la acción.

Tabla 36: Log de acciones -Administrador DJANGO
Fuente: Creación automática DJANGO

viii. LISTA DE MODELOS O TABLAS (TIPOS DE MODELOS) DE LA BASE DE DATOS DEL APLICATIVO

La tabla 37 se crea automáticamente con el framework Django bajo el nombre de django_content_type, la cual contiene la lista de modelos o tables (tipos de modelos) de la base de datos del aplicativo.

Tabla: django_content_type						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
app_label	varchar (100)	NULL		NO		El nombre de la aplicación de la que forma parte el modelo
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
model	varchar (100)	NULL		NO		El nombre de la clase de modelo.

Tabla 37: Modelos de tablas
Fuente: Creación automática DJANGO

ix. CONTROL DE MIGRACIONES (VERSIONES) DE CAMBIOS EN LA BASE DE DATOS

La tabla 38 se crea automáticamente con el framework Django bajo el nombre de `django_migrations`, la cual contiene el control de migraciones (versiones) de cambios en la base de datos.

Tabla: <code>django_migrations</code>						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
<code>app</code>	<code>varchar (255)</code>	NULL		NO		Aplicación que realiza el cambio.
<code>applied</code>	<code>datetime</code>	NULL		NO		Fecha y Hora cuando se aplicó.
<code>id</code>	<code>int (11)</code>	NULL	PRI	NO	<code>auto_increment</code>	Identificador de tuplas de la tabla.
<code>name</code>	<code>varchar (255)</code>	NULL		NO		Nombre del archivo que realiza el cambio.

Tabla 38: Control de migraciones
Fuente: Creación automática DJANGO

x. REGISTRO DEL INICIO DE SESIONES DE LOS USUARIOS EN EL APLICATIVO

La tabla 39 se crea automáticamente con el framework Django bajo el nombre de `django_session`, la cual contiene el registro del inicio de sesiones de los usuarios en el aplicativo.

Tabla: <code>django_session</code>						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
<code>expire_date</code>	<code>datetime</code>	NULL		NO		Una fecha y hora que designa cuándo caduca la sesión.
<code>session_data</code>	<code>longtext</code>	NULL		NO		Una cadena que contiene un diccionario de sesión codificado y serializado.
<code>session_key</code>	<code>varchar (40)</code>	NULL	PRI	NO		Clave primaria. El campo en sí puede contener hasta 40 caracteres.

Tabla 39: Registro de inicio de sesión
Fuente: Creación automática DJANGO

xi. TABLA DEL AVISO

La tabla 40 son creadas por nosotros los tesistas bajo el nombre de main_avisos, la cual contiene todos los avisos.

Tabla: main_avisos						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
cultivo	varchar (100)	NULL		NO		Nombre del cultivo avisado
evaluador_id	int (11)	NULL		YES		Evaluador que atenderá el aviso
f_ajustador	tinyint (1)	NULL		NO		Flag que indicar si el aviso lo está revisando el ajustador
f_desestimado	tinyint (1)	NULL		NO		Flag que indicar si el aviso esta desestimado
f_evaluador	tinyint (1)	NULL		NO		Flag que indicar si el aviso lo está revisando el evaluador
f_fin	tinyint (1)	NULL		NO		Flag que indicar si el aviso ha terminado su proceso
f_indemnizable	tinyint (1)	NULL		NO		Flag que indicar si el aviso es indemnizable
f_minagri	tinyint (1)	NULL		NO		Flag que indicar si el aviso lo está revisando el MINAGRI
f_positiva	tinyint (1)	NULL		NO		Flag que indicar si el aviso lo está revisando Positiva
f_responsable_region	tinyint (1)	NULL		NO		Flag que indicar si el aviso lo está revisando el responsable de región
factor_planta	decimal (15,4)	NULL		YES		Factor calculador por positiva para el cálculo del rendimiento.
fecha_creacion	datetime	NULL		NO		Fecha de creación del aviso
fecha_evaluacion	date	NULL		YES		Fecha de evaluación del aviso
fecha_modificacion	datetime	NULL		NO		Fecha última modificación del aviso
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
minimo_siembra	decimal (15,4)	NULL		YES		Rendimiento que indica cuanto es lo mínimo a producir para no pagar.
monto_pagar	decimal (15,4)	NULL		YES		Monto por pagar, porque ha producido menos que el mínimo de siembra.
pago_por_factor	decimal (15,4)	NULL		YES		Factor calculador por positiva para el cálculo del rendimiento.
sector_estadistico_id	int (11)	NULL		NO		Sector estadístico del aviso.
tipo_evento	varchar (100)	NULL		YES		Tipo evento del aviso.
total_siembra	decimal (15,4)	NULL		YES		Total, siembra del aviso.
usuario_id	int (11)	NULL		NO		Usuario id del aviso.

Tabla 40: Datos de los avisos ingresados

Fuente: Creación propia

xii. COMENTARIOS DEL AVISO

La tabla 41 son creadas por nosotros los tesistas, bajo el nombre de main_avisocomentario, la cual contiene todos los comentarios de los avisos.

Tabla: main_avisocomentario						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
aviso_id	int (11)	NULL		NO		Aviso al que pertenece el comentario.
comentario	varchar (10240)	NULL		NO		Comentario al que pertenece el Aviso.
fecha_creacion	datetime	NULL		NO		Fecha de creación al que pertenece el comentario.
fecha_modificacion	datetime	NULL		NO		Fecha de modificación al que pertenece el comentario.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
usuario_id	int (11)	NULL		NO		Usuario que realizo el comentario.

Tabla 41: Comentarios de los usuarios en los avisos
Fuente: Creación propia

xiii. TABLA DE DEPARTAMENTOS

La tabla 42 son creadas por nosotros los tesistas, bajo el nombre de main_departamento, la cual contiene todos los departamentos beneficiados por el seguro agrario catastrófico.

Tabla: main_departamento						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
fecha_creacion	datetime	NULL		NO		Fecha de creación del registro.
fecha_modificacion	datetime	NULL		NO		Fecha de modificación del registro.
grupo_id	int (11)	NULL		YES		Cada departamento será referenciado por grupos para su asignación de permisos.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
nombre	varchar (50)	NULL		NO		Nombre del registro de la tabla.

Tabla 42: Tabla de Departamentos
Fuente: Creación propia

xiv. TABLA DE PROVINCIAS

La tabla 43 son creadas por nosotros los tesistas, bajo el nombre de main_provincia, la cual contiene todas las provincias beneficiados por el seguro agrario catastrófico.

Tabla: main_provincia						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
departamento_id	int (11)	NULL		YES		Departamento al que pertenece la provincia.
fecha_creacion	datetime	NULL		NO		Fecha de creación del registro.
fecha_modificacion	datetime	NULL		NO		Fecha de modificación del registro.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
nombre	varchar (50)	NULL		NO		Nombre del registro de la tabla.

Tabla 43: Tabla de provincias
Fuente: Creación propia

xv. TABLA DE DISTRITOS

La tabla 44 son creadas por nosotros los tesistas, bajo el nombre de main_distrito, la cual contiene todos los distritos beneficiados por el seguro agrario catastrófico.

Tabla: main_distrito						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
fecha_creacion	datetime	NULL		NO		Fecha de creación del registro.
fecha_modificacion	datetime	NULL		NO		Fecha de modificación del registro.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
nombre	varchar (50)	NULL		NO		Nombre del registro de la tabla.
provincia_id	int (11)	NULL		YES		Provincia al que pertenece el distrito.

Tabla 44: Tabla de distritos
Fuente: Creación propia

xvi. TABLA DE SECTOR ESTADÍSTICO

La tabla 45 son creadas por nosotros los tesistas, bajo el nombre de main_sectorestadistico, la cual contiene todos los sectores estadísticos beneficiados por el seguro agrario catastrófico.

Tabla: main_sectorestadistico						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
distrito_id	int (11)	NULL		YES		Distrito al que pertenece el sector estadístico.
fecha_creacion	datetime	NULL		NO		Fecha de creación del registro.
fecha_modificacion	datetime	NULL		NO		Fecha de modificación del registro.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
nombre	varchar (50)	NULL		NO		Nombre del registro de la tabla.
ogr_fid	int (11)	NULL		YES		Identificador de figura.

Tabla 45: Tabla de sector estadístico
Fuente: Creación propia

xvii. LOG DE LOGINS

La tabla 46 son creadas por nosotros los tesistas, bajo el nombre de main_loglogin, la cual contiene un registro de ingreso inicios de sesión exitosos y fallidos.

Tabla: main_loglogin						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
fecha_creacion	datetime	NULL		NO		Fecha de creación del registro.
fecha_modificacion	datetime	NULL		NO		Fecha de modificación del registro.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
remote_addr	varchar (50)	NULL		YES		Dirección IP del Login
usuario_id	int (11)	NULL		YES		Usuario que inicia el Login
usuario_texto	varchar (50)	NULL		YES		Mensaje del Login (Correcto, Incorrecto)

Tabla 46: Control de logins
Fuente: Creación propia

xviii. GENERAR PUNTOS Y MUESTRA DE EVALUACIÓN DE LOS AVISOS

La tabla 47 son creadas por nosotros los tesistas, bajo el nombre de main_punto, la cual contiene los puntos de evaluación.

Tabla: main_punto						
Columna	Tipo	Default	Key	Nulo	Auto Increment	Comentario
aviso_id	int (11)	NULL		NO		Aviso al que pertenece el punto
calculo	decimal (15,4)	NULL		YES		Cálculo del rendimiento del punto numero_plantas x cultivo_plantas
cultivo_plantas	decimal (15,4)	NULL		YES		Valor de cultivo por planta del punto
este	varchar (20)	NULL		YES		Coordenada este del punto.
f_registra_analisis	tinyint (1)	NULL		NO		Verifica si requiere análisis el punto,
fecha_creacion	datetime	NULL		NO		Fecha de creación del registro.
fecha_modificacion	datetime	NULL		NO		Fecha de modificación del registro.
id	int (11)	NULL	PRI	NO	auto_increment	Identificador de tuplas de la tabla.
norte	varchar (20)	NULL		YES		Coordenada norte del punto.
numero	int (11)	NULL		NO		Número de punto.
numero_plantas	decimal (15,4)	NULL		YES		Número de plantas en el punto
siembra	decimal (15,4)	NULL		YES		Valor de siembra del punto = calculo x Aviso.factor_planta

Tabla 47: Generar puntos y muestra de evaluación de los avisos
Fuente: Creación propia

B. TARJETAS CRC

- En la tarjeta CRC de la clase Auth_User, a continuación, podemos verificar a través de la siguiente tabla 48 la colaboración que existe:

Clase: Auth_User	
Responsabilidades	Colaboradores
1. Registrar los usuarios del aplicativo.	

Tabla 48: CRC Auth_User
Fuente: Creación automática DJANGO

- En la tarjeta CRC de la clase Auth_Group, a continuación, podemos verificar a través de la siguiente tabla 49 la colaboración que existe:

Clase: Auth_Group	
Responsabilidades	Colaboradores
1. Registrar los grupos del aplicativo. (Roles)	

Tabla 49: CRC Auth_Group
Fuente: Creación automática DJANGO

- En la tarjeta CRC de la clase LogLogin, a continuación, podemos verificar a través de la siguiente tabla 50 la colaboración que existe:

Clase: LogLogin	
Responsabilidades	Colaboración
1. Guardar Información de los logins de los usuarios por ips 2. Guardar Información de los logins fallidos 3. Registrar por usuarios lo ingresos.	

Tabla 50: CRC LogLogin
Fuente: Creación propia

- En la tarjeta CRC de la clase Departamento, a continuación, podemos verificar a través de la siguiente tabla 51 la colaboración que existe:

Clase: Departamento	
Responsabilidades	Colaboradores
1. CRUD los departamentos con los que se trabajaran. 2. Relacionar los departamentos a cada grupo de usuarios de la aplicación.	

Tabla 51: CRC Departamento
Fuente: Creación propia

- En la tarjeta CRC de la clase Provincia, a continuación, podemos verificar a través de la siguiente tabla 52 la colaboración que existe:

Clase: Provincia	
Responsabilidades	Colaboradores
1. CRUD las Provincias con los que se trabajaran asociadas a un departamento.	Departamento

Tabla 52: CRC Provincia
Fuente: Creación propia

- En la tarjeta CRC de la clase Distrito, a continuación, podemos verificar a través de la siguiente tabla 53 la colaboración que existe:

Clase: Distrito	
Responsabilidades	Colaboradores
1. CRUD las Distrito con los que se trabajaran asociadas a una provincia.	Provincia

Tabla 53: CRC Distrito
Fuente: Creación propia

- En la tarjeta CRC de la clase SectorEstadistico, a continuación, podemos verificar a través de la siguiente tabla 54 la colaboración que existe:

Clase: SectorEstadistico	
Responsabilidades	Colaboradores
1. CRUD los Sectores Estadísticos con los que se trabajaran asociadas a un distrito.	Distrito

Tabla 54: CRC Sector estadístico
Fuente: Creación propia

- En la tarjeta CRC de la clase Cultivo, a continuación, podemos verificar a través de la siguiente tabla 55 la colaboración que existe:

Clase: Cultivo	
Responsabilidades	Colaboradores
1. CRUD los cultivos con los que se trabajaran.	

Tabla 55: CRC Cultivo
Fuente: Creación propia

- En la tarjeta CRC de la clase Aviso, a continuación, podemos verificar a través de la siguiente tabla 56 la colaboración que existe:

Clase: Aviso	
Responsabilidades	Colaboradores
1. Registrar los Aviso con los que se trabajaran asociadas a un distrito. 2. Cambiar de estados el aviso (RESPONSABLE DE REGIÓN, MINAGRI, POSITIVA, AJUSTADOR, EVALUADOR, FINALIZADO) 3. Asignar Evaluador. 4. Evaluar aviso. 5. Desestimar	SectorEstadistico Auth_User Cultivo

Tabla 56: CRC Aviso
Fuente: Creación propia

- En la tarjeta CRC de la clase Punto, a continuación, podemos verificar a través de la siguiente tabla 57 la colaboración que existe:

Clase: Punto	
Responsabilidades	Colaboradores
1. CRUD los datos de evaluación de cada punto.	Aviso

Tabla 57: CRC Punto de evaluación
Fuente: Creación propia

- En la tarjeta CRC de la clase AvisoLog, a continuación, podemos verificar a través de la siguiente tabla 58 la colaboración que existe:

Clase: AvisoLog	
Responsabilidades	Colaboradores
1. Registrar los cambios de estados con fecha, hora y quien lo realiza.	Aviso Auth_User

Tabla 58: CRC Registrar cambios de estado de los avisos
Fuente: Creación propia

- En la tarjeta CRC de la clase AvisoComentario, a continuación, podemos verificar a través de la siguiente tabla 59 la colaboración que existe:

Clase: AvisoComentario	
Responsabilidades	Colaboradores
1. Registrar comentarios en el aviso por usuario.	Aviso Auth_User

Tabla 59: CRC Registrar comentarios de los avisos
Fuente: Creación propia

III. CODIFICACIÓN DEL PROYECTO

A. ITERACIÓN I

a. HU N°1: INTEGRAR CARTOGRAFÍA CON SECTORES ESTADÍSTICOS

i. Modelos o Tablas:

En la imagen 40 podemos visualizar los modelos a utilizar como son departamentos, provincia, distrito y sector estadísticos.

```
20 class Departamento(models.Model):
21     nombre=models.CharField(max_length=50)
22     grupo=models.ForeignKey(Group,null=True,blank=True)
23
24     fecha_creacion = models.DateTimeField(auto_now_add=True)
25     fecha_modificacion = models.DateTimeField(auto_now=True)
26     def __unicode__(self):
27         return (unicode) (self.nombre)
28
29 class Provincia(models.Model):
30     nombre=models.CharField(max_length=50)
31     departamento=models.ForeignKey("Departamento",null=True,blank=True)
32
33     fecha_creacion = models.DateTimeField(auto_now_add=True)
34     fecha_modificacion = models.DateTimeField(auto_now=True)
35     def __unicode__(self):
36         return (unicode) (self.nombre)
37
38 class Distrito(models.Model):
39     nombre=models.CharField(max_length=50)
40     provincia=models.ForeignKey("Provincia",null=True,blank=True)
41
42     fecha_creacion = models.DateTimeField(auto_now_add=True)
43     fecha_modificacion = models.DateTimeField(auto_now=True)
44     def __unicode__(self):
45         return (unicode) (self.nombre)
46
47 class SectorEstadistico(models.Model):
48     nombre=models.CharField(max_length=50)
49     distrito=models.ForeignKey("Distrito",null=True,blank=True)
50     ogr_fid=models.IntegerField(null=True,blank=True)
51
52     fecha_creacion = models.DateTimeField(auto_now_add=True)
53     fecha_modificacion = models.DateTimeField(auto_now=True)
```

Imagen 40: Codificación modelos/ tablas a usar en la HU N°1

Fuente: Creación propia

ii. Proceso:

En la imagen 41 visualizamos el proceso principal que devuelve la URL del mapa junto a los puntos de evaluación la cual el usuario final consumirá.

```
157 def get_url_mapa(se):
158     d = ogr.GetDriverByName("MySQL")
159     ds = d.Open("MYSQL:shapes_minagri_4326,user=root", 0)
160     l=ds.GetLayerByName("shape_sectores_minagri")
161     l.SetAttributeFilter("OGR_FID = %s" % se.ogr_fid)
162     f=l.GetNextFeature()
163     g=f.GetGeometryRef()
164     e=g.GetEnvelope()
165     x=[e[0], e[2], e[1], e[3]]
166     m=mapscript.mapObj("./cartografia/base/01.map")
167     m.setExtent(x[0],x[1],x[2],x[3])
168     l_y(m, "./cartografia/base/02.map")
169     l_y(m, "./cartografia/base/03.map", dict_replaces={"####OGR_FID####":se.ogr_fid})
170     l_y(m, "./cartografia/base/04.map")
171
172     l_y(m, "./cartografia/base/mry_02.map")
173     l_y(m, "./cartografia/base/mry_03.map")
174     l_y(m, "./cartografia/base/mry_04.map")
175     l_y(m, "./cartografia/base/mry_05.map")
176     l_y(m, "./cartografia/base/mry_09.map")
177     l_y(m, "./cartografia/base/mry_14.map")
178     l_y(m, "./cartografia/base/mry_17.map")
179     l_y(m, "./cartografia/base/mry_18.map")
180     l_y(m, "./cartografia/base/mry_20.map")
181
182     ps=get_punto_ale(m,g,x)
183
184     rps=[]
185     for p in ps:
186         rps.append({'numero':p.numero,'este':p.este,'norte':p.norte})
187
188     i=m.draw()
189     i.save("./static/imgcart/mp/%s.png" % se.id )
190     return {
191         'url':static("imgcart/mp/%s.png" % se.id ),
192         'puntos':rps
193     }
194
```

Imagen 41: Codificación del proceso principal para generar el mapa de la HUN°1
Fuente: Creación propia

iii. Resultado:

En la imagen 42 podemos observar el resultado final, el cual obtendremos el mapa con los puntos de evaluación.



Imagen 42: Resultado de la Codificación de la HU N°1
Fuente: Creación propia

B. ITERACIÓN II

a. HU N°2: REGISTRO DE AVISO

i. Modelos o tablas:

En la imagen 43 observamos los modelos por utilizar, los cuales son aviso, sector estadístico, cultivo, tipo de evento, auth user.

```
56 class Aviso(models.Model):
57
58     CULTIVOS_CHOICES = (
59         ('PAPA', 'PAPA'),
60         ('MAIZ AMILACIO', 'MAIZ AMILACIO'),
61         ('ARVERJA ', 'ARVERJA'),
62         ('QUINUA ', 'QUINUA'),
63         ('YUCA', 'YUCA'),
64     )
65
66
67     TIPO_EVENTO_CHOICES = (
68         ('SEQUIA', 'SEQUIA'),
69         ('INUNDACIÓN', 'INUNDACIÓN'),
70         ('HUAYCO', 'HUAYCO'),
71         ('DESGLIZAMIENTO', 'DESGLIZAMIENTO'),
72         ('TEMPERATURAS EXTREMAS', 'TEMPERATURAS EXTREMAS'),
73         ('GRANIZOS', 'GRANIZOS'),
74         ('VIENTOS FUERTES', 'VIENTOS FUERTES'),
75     )
76
77     sector_estadistico=models.ForeignKey("SectorEstadistico")
78
79     cultivo=models.CharField(max_length=100,choices=CULTIVOS_CHOICES)
80     tipo_evento=models.CharField(max_length=100,choices=TIPO_EVENTO_CHOICES,null=True,blank=True)
81
82     factor_planta=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
83     total_siembra=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
84     minimo_siembra=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
85     pago_por_factor=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#si mi
86
87     f_indemnizable=models.BooleanField(default=False)
88     monto_pagar=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)
89
90     f_responsable_region=models.BooleanField(default=False)
91     f_minagri=models.BooleanField(default=False)
92     f_positiva=models.BooleanField(default=False)
93     f_ajustador=models.BooleanField(default=False)
94     f_evaluador=models.BooleanField(default=False)
95     f_fin=models.BooleanField(default=False)
96
97
98     f_desestimado=models.BooleanField(default=False)
99
100     evaluador=models.ForeignKey(User,null=True,blank=True,related_name="evaluador")
101
102
103     usuario=models.ForeignKey(User)
104
105     fecha_evaluacion = models.DateField(null=True,blank=True)
106
107     fecha_creacion = models.DateTimeField(auto_now_add=True)
108     fecha_modificacion = models.DateTimeField(auto_now=True)
109
```

Imagen 43: Codificación modelos/ tablas a usar en la HU N°2
Fuente: Creación propia

ii. Proceso:

En la imagen 44 mostramos el proceso principal del registro del aviso.

```
97
98 def registrar_aviso(request):
99     if fnc.session.valida(request):
100         registro_ok=False
101         if request.POST.has_key("sector_estadistico") and
102            request.POST.has_key("cultivo") and
103            request.POST.has_key("txt_puntos") and
104            request.POST.has_key("tipo_evento"):
105
106             txt_puntos=request.POST["txt_puntos"].strip()
107             txt_ps=txt_puntos.split('||||')
108             correcto=True
109             puntos=[]
110             for ip in range(0,len(txt_ps)-1):
111                 txt_p=txt_ps[ip]
112                 arr=txt_p.split('****')
113                 if len(arr)==3:
114                     puntos.append(Punto(numero=arr[0],este=arr[1],norte=arr[2]))
115                 else:
116                     correcto=False
117
118             if correcto:
119                 aviso=Aviso.objects.create(
120                     sector_estadistico_id=request.POST["sector_estadistico"],
121                     cultivo=request.POST["cultivo"],
122                     tipo_evento=request.POST["tipo_evento"],
123                     usuario=request.user,
124                     f_minagri=True
125                 )
126
127                 for punto in puntos:
128                     punto.aviso=aviso
129                     punto.save()
130
131                 mapa.get_url_mapa_aviso(aviso)
132
133                 registro_ok=True
134
135             return fnc.render.run(request,"main/registrar_aviso.html",{
136                 "registro_ok":registro_ok,
137                 "cultivos":Aviso.CULTIVOS_CHOICES,
138                 "tipos_eventos":Aviso.TIPO_EVENTO_CHOICES
139             })
140     return fnc.redireccionar.login()
141
```

Imagen 44: Codificación del proceso principal para registrar el aviso de la HUN^o2
Fuente: Creación propia

iii. Resultado:

El resultado de la codificación de la HU N^o2, la mostramos a continuación en la imagen 45:

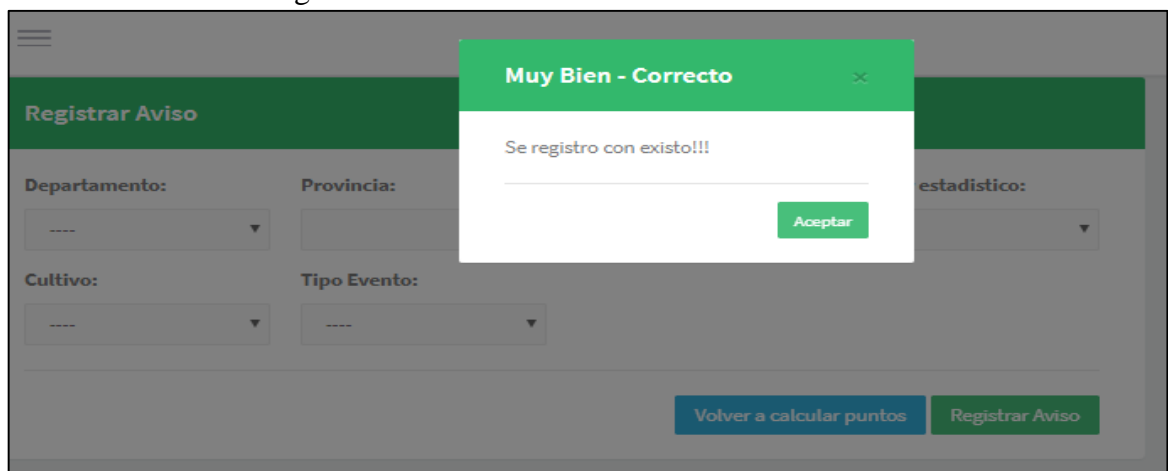
The image shows a web application interface for registering an alert. A green modal box with the title 'Muy Bien - Correcto' is centered on the screen, displaying the message 'Se registro con existo!!!' (Note the typo 'existo' instead of 'exito'). Below the message is a green 'Aceptar' button. In the background, the 'Registrar Aviso' form is visible, featuring dropdown menus for 'Departamento', 'Provincia', 'Cultivo', and 'Tipo Evento', and a 'sector_estadistico' dropdown. At the bottom of the form are two buttons: 'Volver a calcular puntos' and 'Registrar Aviso'.

Imagen 45: Resultado de la Codificación de la HU N^o2
Fuente: Creación propia

b. HU N° 3: GENERAR MAPA DEL SECTOR ESTADÍSTICO CON PUNTOS A EVALUAR – WEB

i. Modelos o Tablas:

Cómo se muestra en la imagen 46, observamos que los modelos por utilizar son aviso, sector estadístico

```
47 class SectorEstadistico(models.Model):
48     nombre=models.CharField(max_length=50)
49     distrito=models.ForeignKey("Distrito",null=True,blank=True)
50     ogr_fid=models.IntegerField(null=True,blank=True)
51
52     fecha_creacion = models.DateTimeField(auto_now_add=True)
53     fecha_modificacion = models.DateTimeField(auto_now=True)
54
55
56 class Aviso(models.Model):
57
58     CULTIVOS_CHOICES = (
59         ('PAPA', 'PAPA'),
60         ('MAIZ AMILACIO', 'MAIZ AMILACIO'),
61         ('ARVERJA ', 'ARVERJA'),
62         ('QUINUA ', 'QUINUA'),
63         ('YUCA', 'YUCA'),
64     )
65
66
67     TIPO_EVENTO_CHOICES = (
68         ('SEQUÍA', 'SEQUÍA'),
69         ('INUNDACIÓN', 'INUNDACIÓN'),
70         ('HUAYCO', 'HUAYCO'),
71         ('DESGLIZAMIENTO', 'DESGLIZAMIENTO'),
72         ('TEMPERATURAS EXTREMAS', 'TEMPERATURAS EXTREMAS'),
73         ('GRANIZOS', 'GRANIZOS'),
74         ('VIENTOS FUERTES', 'VIENTOS FUERTES'),
75     )
76
77     sector_estadistico=models.ForeignKey("SectorEstadistico")
78
79     cultivo=models.CharField(max_length=100,choices=CULTIVOS_CHOICES)
80     tipo_evento=models.CharField(max_length=100,choices=TIPO_EVENTO_CHOICES,null=True,blank=True)
81
82     factor_planta=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
83     total_siembra=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
84     minimo_siembra=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
85     pago_por_factor=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#si mi
86
87     f_indemnizable=models.BooleanField(default=False)
88     monto_pagar=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)
89
90     f_responsable_region=models.BooleanField(default=False)
91     f_minagri=models.BooleanField(default=False)
92     f_positiva=models.BooleanField(default=False)
93     f_ajustador=models.BooleanField(default=False)
94     f_evaluador=models.BooleanField(default=False)
95     f_fin=models.BooleanField(default=False)
96
97
98     f_desestimado=models.BooleanField(default=False)
99
100     evaluador=models.ForeignKey(User,null=True,blank=True,related_name="evaluador")
101
102
103     usuario=models.ForeignKey(User)
104
105     fecha_evaluacion = models.DateField(null=True,blank=True)
106
107     fecha_creacion = models.DateTimeField(auto_now_add=True)
108     fecha_modificacion = models.DateTimeField(auto_now=True)
109
```

Imagen 46: Codificación modelos/ tablas a usar en la HU N°3

Fuente: Creación propia

ii. Proceso:

En la imagen 47 mostramos el proceso principal de la generación de los puntos de evaluación.

```
197 se=av.sector_estadistico
198 d = ogr.GetDriverByName("MySQL")
199 ds = d.Open("MySQL:shapes_minagri_4326,user=root", 0)
200 l=ds.GetLayerByName("shape_sectores_minagri")
201 l.SetAttributeFilter("OGR_FID = %s" % se.ogr_fid)
202 f=l.GetNextFeature()
203 g=f.GetGeometryRef()
204 e=g.GetEnvelope()
205 x=[e[0], e[2], e[1], e[3]]
206 m=mapscript.mapObj("./cartografia/base/01.map")
207 m.setExtent(x[0],x[1],x[2],x[3])
208 l_y(m,"./cartografia/base/02.map")
209 l_y(m,"./cartografia/base/03.map",dict_replaces={"####OGR_FID####":se.ogr_fid})
210 l_y(m,"./cartografia/base/04.map")
211
212 l_y(m,"./cartografia/base/mry_02.map")
213 l_y(m,"./cartografia/base/mry_03.map")
214 l_y(m,"./cartografia/base/mry_04.map")
215 l_y(m,"./cartografia/base/mry_05.map")
216 l_y(m,"./cartografia/base/mry_09.map")
217 l_y(m,"./cartografia/base/mry_14.map")
218 l_y(m,"./cartografia/base/mry_17.map")
219 l_y(m,"./cartografia/base/mry_18.map")
220 l_y(m,"./cartografia/base/mry_20.map")
221
222 for p in av.punto_set.all():
223     if p.este!=None and p.norte!=None and p.este!="" and p.norte!="":
224         a=get_dec(p.este)
225         b=get_dec(p.norte)
226         if a!=None and b!=None:
227             l_y(m,"./cartografia/base/fi_01.map",dict_replaces={"####a####":a,"####b####":b})
228
229 i=m.draw()
230 i.save("./static/imgcart/ma/%s.png" % av.id )
231 return static("imgcart/ma/%s.png" % av.id )
```

Imagen 47: Codificación del proceso principal para generar mapa y puntos en la web de la HUN^o3
Fuente: Creación propia

iii. Resultado:

En la imagen 48, mostramos el resultado de nuestra codificación por la HU N^o3

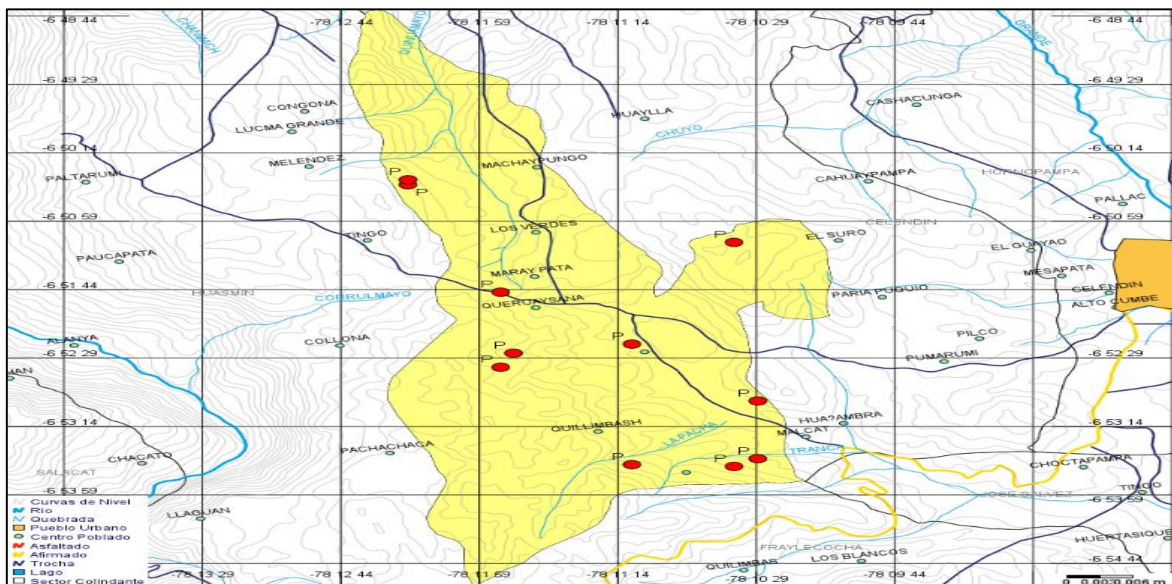


Imagen 48: Resultado de la Codificación de la HU N^o3
Fuente: Creación propia

c. HU N° 4: MOSTRAR MAPA DEL SECTOR ESTADÍSTICO – TABLET

i. Modelos o Tablas:

Em la imagen 49 mostramos los modelos por utilizar los cuales son aviso, sector estadístico.

```
47 class SectorEstadistico(models.Model):
48     nombre=models.CharField(max_length=50)
49     distrito=models.ForeignKey("Distrito",null=True,blank=True)
50     ogr_fid=models.IntegerField(null=True,blank=True)
51
52     fecha_creacion = models.DateTimeField(auto_now_add=True)
53     fecha_modificacion = models.DateTimeField(auto_now=True)
54
55
56 class Aviso(models.Model):
57
58     CULTIVOS_CHOICES = (
59         ('PAPA', 'PAPA'),
60         ('MAIZ AMILACIO', 'MAIZ AMILACIO'),
61         ('ARVERJA ', 'ARVERJA'),
62         ('QUINUA ', 'QUINUA'),
63         ('YUCA', 'YUCA'),
64     )
65
66
67     TIPO_EVENTO_CHOICES = (
68         ('SEQUIA', 'SEQUIA'),
69         ('INUNDACIÓN', 'INUNDACIÓN'),
70         ('HUAYCO', 'HUAYCO'),
71         ('DESLIZAMIENTO', 'DESLIZAMIENTO'),
72         ('TEMPERATURAS EXTREMAS', 'TEMPERATURAS EXTREMAS'),
73         ('GRANIZOS', 'GRANIZOS'),
74         ('VIENTOS FUERTES', 'VIENTOS FUERTES'),
75     )
76
77     sector_estadistico=models.ForeignKey("SectorEstadistico")
78
79     cultivo=models.CharField(max_length=100,choices=CULTIVOS_CHOICES)
80     tipo_evento=models.CharField(max_length=100,choices=TIPO_EVENTO_CHOICES,null=True,blank=True)
81
82     factor_planta=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
83     total_siembra=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
84     minimo_siembra=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#en Kg
85     pago_por_factor=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)#si mi
86
87     f_indemnizable=models.BooleanField(default=False)
88     monto_pagar=models.DecimalField(max_digits=15, decimal_places=4,null=True,blank=True)
89
90     f_responsable_region=models.BooleanField(default=False)
91     f_minagri=models.BooleanField(default=False)
92     f_positiva=models.BooleanField(default=False)
93     f_ajustador=models.BooleanField(default=False)
94     f_evaluador=models.BooleanField(default=False)
95     f_fin=models.BooleanField(default=False)
96
97
98     f_desestimado=models.BooleanField(default=False)
99
100     evaluador=models.ForeignKey(User,null=True,blank=True,related_name="evaluador")
101
102
103     usuario=models.ForeignKey(User)
104
105     fecha_evaluacion = models.DateField(null=True,blank=True)
106
107     fecha_creacion = models.DateTimeField(auto_now_add=True)
108     fecha_modificacion = models.DateTimeField(auto_now=True)
109
```

Imagen 49: Codificación modelos/ tablas a usar en la HU N°4

Fuente: Creación propia

ii. Proceso:

En la imagen 50 mostramos el proceso principal para envío de los mapas hacia la Tablet.

```
89
90 @csrf_exempt
91 def descargar_pendientes(request):
92     json=get_json_default()
93     if request.POST.has_key("i") :
94         evaluadores=User.objects.filter(groups__name=settings.GRUPO_EVALUADOR,id=request.POST["i"])
95         if evaluadores.count()==1:
96             avisos=Aviso.objects.filter(evaluador_id=request.POST["i"],f_evaluador=True)
97
98             data=[]
99
100             for aviso in avisos:
101                 ps=[]
102                 for p in aviso.punto_set.all():
103                     ps.append({"nu":p.numero,"es":p.este,"no":p.norte})
104                 av={
105                     "i":aviso.id,
106                     "de":aviso.sector_estadistico.districto.provincia.departamento.nombre,
107                     "pr":aviso.sector_estadistico.districto.provincia.nombre,
108                     "di":aviso.sector_estadistico.districto.nombre,
109                     "se":aviso.sector_estadistico.nombre,
110                     "cu":aviso.cultivo,
111                     "te":aviso.tipo_evento,
112                     "fa": "%s" % aviso.factor_planta,
113                     "ur": "%s" % aviso.get_url_mapa_aviso(),
114                     "ps":ps,
115                 }
116                 data.append(av)
117
118             json["data"]=data
119             json["success"]=1
120
121             return get_response(json)
122
```

Imagen 50: Codificación del proceso principal para envía URL de mapas desde la web hasta las tablets de la HUN°4

Fuente: Creación propia

En la imagen 51 mostramos el proceso principal para descargar la URL del mapa en la tablet

```
154
155
156 private static void descargar_imagenes(Map<Long,String> urls_imagenes){
157     for(final Map.Entry <Long,String> entry : urls_imagenes.entrySet()){
158         AsyncTask asyncTask = new AsyncTask() {
159
160             @Override
161             protected Object doInBackground(Object[] params) {
162                 try {
163                     URL url = new URL( spec HOST + entry.getValue());
164                     URLConnection conn = url.openConnection();
165
166                     InputStream input = conn.getInputStream();
167                     OutputStream output = new FileOutputStream(fnc.getPathImagen(entry.getKey()));
168
169                     int fileLength = conn.getContentLength();
170
171                     byte data[] = new byte[4096];
172                     long total = 0;
173                     int count;
174                     while ((count = input.read(data)) != -1) {
175                         if (isCancelled()) {
176                             input.close();
177                             return null;
178                         }
179                         total += count;
180                         if (fileLength > 0) publishProgress((int) (total * 100 / fileLength));
181                         output.write(data, 0, count);
182                     }
183
184                     if (output != null) output.close();
185                     if (input != null) input.close();
186                 } catch (Exception ex) {
187                     ex.printStackTrace();
188                 }
189                 return null;
190             }
191         }.execute();
192     }
193 }
```

Imagen 51: Proceso principal para descargar la URL del mapa en la Tablet de la HUN°4

Fuente: Creación propia

iii. Resultados:

El resultado de nuestra codificación queda expresado en la imagen 52 que mostramos a continuación:



Imagen 52: Resultado de la Codificación de la HU N°4
Fuente: Creación propia

d. HU N° 5: ADMINISTRAR BANDEJA DE AVISOS

i. Proceso:

En la imagen 53 mostramos la codificación del proceso principal para Administrar Bandejas de avisos en la web.

```
128 def listar_avisos(request):
129     if fnc.session.valida(request):
130         txt_filtro=None
131         estado_filtro="todos"
132         if request.GET and request.GET.has_key("txt_filtro"):
133             txt_filtro=request.GET["txt_filtro"].strip()
134
135
136         if request.GET and request.GET.has_key("estado_filtro"):
137             estado_filtro=request.GET["estado_filtro"].lower()
138             if not estado_filtro in ["responsable_region", "minagri", "positiva", "ajustador", "evaluador", "fin"]:
139                 estado_filtro="todos"
140         avisos=Aviso.objects.all()
141         if estado_filtro!="todos":
142             if estado_filtro=="responsable_region":
143                 avisos=avisos.filter(f_responsable_region=True)
144             elif estado_filtro=="minagri":
145                 avisos=avisos.filter(f_minagri=True)
146             elif estado_filtro=="positiva":
147                 avisos=avisos.filter(f_positiva=True)
148             elif estado_filtro=="ajustador":
149                 avisos=avisos.filter(f_ajustador=True)
150             elif estado_filtro=="evaluador":
151                 avisos=avisos.filter(f_evaluador=True)
152             elif estado_filtro=="fin":
153                 avisos=avisos.filter(f_fin=True)
154
155         if txt_filtro!=None:
156             avisos=avisos.filter(Q(cultivo__icontains=txt_filtro) | Q(tipo_evento__icontains=txt_filtro) | Q(se
157
158     return fnc.render.run(request,"main/listar_avisos.html",{ "avisos":avisos,"estado_filtro":estado_filtro,
159     return fnc.redireccionar.login()
```

Imagen 53:Codificación del proceso principal para Administrar Bandejas de avisos en la web de la HUN°5

Fuente: Creación propia

ii. Resultado:

En la imagen 54 mostramos el resultado de nuestra codificación para el proceso principal de Administrar Bandejas de avisos en la web.

Listar Aviso (21)									
Todos		Buscar							
Cultivo	Responsable de región	Titulo	Provincia	Sector	Evento	Creado	Resultado	Estado	Detalle
PAPA	Minagri	IN	CHONTALI	PICHAPIRIANA	SEQUÍA	Mar. 13, 2018, 12:48 p.m.	En Proceso	Contador	Ver Detalle
MAIZ AMILACIO	Positiva	CAJAMARCA	SAN PABLO	SAN LUIS	PAMPYA SAN LUIS	INUNDACIÓN	Mar. 13, 2018, 12:52 p.m.	En Proceso	Responsable Region
MAIZ AMILACIO	Ajustador	CAJAMARCA	JAEN	SALLIQUE	CHALANBANCHE	GRANIZOS	Mar. 13, 2018, 12:52 p.m.	En Proceso	Contador
MAIZ AMILACIO	Evaluador	CAJAMARCA	SAN MIGUEL	SAN GREGORIO	SAN GREGORIO	INUNDACIÓN	Mar. 13, 2018, 12:52 p.m.	En Proceso	Ajustador
MAIZ AMILACIO	Terminados	CAJAMARCA	SAN PABLO	SAN PABLO	JANCOS ALTO	TEMPERATURAS EXTREMAS	Mar. 13, 2018, 12:53 p.m.	En Proceso	Minagri
MAIZ AMILACIO		CAJAMARCA	SAN PABLO	SAN PABLO	CALLANCA	DESPLAZAMIENTO	Mar. 13, 2018, 12:53 p.m.	En Proceso	Minagri
PAPA		CAJAMARCA	CUTERVO	SAN LUIS DE LUCHA	SAIREPAMPA	HUAYCO	Mar. 13, 2018, 12:53 p.m.	En Proceso	Positiva
ARVERJA		CAJAMARCA	SAN MIGUEL	EL PRADO	EL PRADO	DESPLAZAMIENTO	Mar. 13, 2018, 12:53 p.m.	En Proceso	Contador

Imagen 54:Resultado de la Codificación de la HU N°5

Fuente: Creación propia

e. HU N° 6: DESCARGAR AVISOS EN TABLET

i. Proceso:

En la imagen 55 mostramos la codificación del proceso principal para descargar avisos en la Tablet

```

194 public static Integer descargar_pendientes(){
195     if (fnc.USUARIO_SID!=null){
196         Map<Long,String> urls_imagenes=new HashMap<>();
197         AsyncTask asyncTask = (params) -> {
201             try {
202                 String data = URLEncoder.encode("i", "UTF-8") + "=" + URLEncoder.encode(fnc.USUARIO_SID+"", "UTF-8");
203
204                 URL url = new URL( spec: HOST + "ajax/descargar_pendientes/");
205                 URLConnection conn = url.openConnection();
206                 conn.setDoOutput(true);
207                 OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());
208                 wr.write(data);
209                 wr.flush();
210
211
212
213                 BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
214                 StringBuilder sb = new StringBuilder();
215                 String line = null;
216                 while ((line = reader.readLine()) != null) {
217                     sb.append(line + "\n");
218                 }
219                 String text = sb.toString();
220                 return text;
221             } catch (Exception ex) {
222                 ex.printStackTrace();
223             }
224             return null;
225         };
226         String respuesta = null;

```

Imagen 55:Codificación del proceso principal para descargar avisos en la Tablet de la HUN°6

Fuente: Creación propia

ii. Resultado:

En la imagen 56 mostramos el resultado de la codificación del proceso principal para descargar avisos en la Tablet

TesisApp							
DESCARGAR DATOS CAMAYA		Todos		Buscar...		BUSCAR	
N°	Codigo	Cultivo	Departamento	Provincia	Distrito	Sector	Evento
1	1	PAPA	CAJAMARCA	JAEN	CHONTALI	PACHAPIRIANA	SEQUÍA
2	2	ARVERJA	CAJAMARCA	SAN MIGUEL	EL PRADO	EL PRADO	DESLIZAM
3	3	QUINUA	CAJAMARCA	SAN MARCOS	EDUARDO VILLANUEVA	HUACACORRAL	TEMPERA
4	4	MAIZ AMILACIO	CAJAMARCA	SAN PABLO	TUMBADEN	INGATAMBO	HUAYCO
5	5	PAPA	CAJAMARCA	CAJABAMBA	CAJABAMBA	CAJABAMBA	INUNDAC

Imagen 56:Resultado de la Codificación de la HU N°6

Fuente: Creación propia

C. ITERACIÓN III

a. HU N° 7: REGISTRAR DATOS DE SINIESTROS POR AVISO – TABLET

i. Proceso:

En la imagen 57 mostramos la codificación del proceso principal para registrar datos de siniestro por aviso en la Tablet.

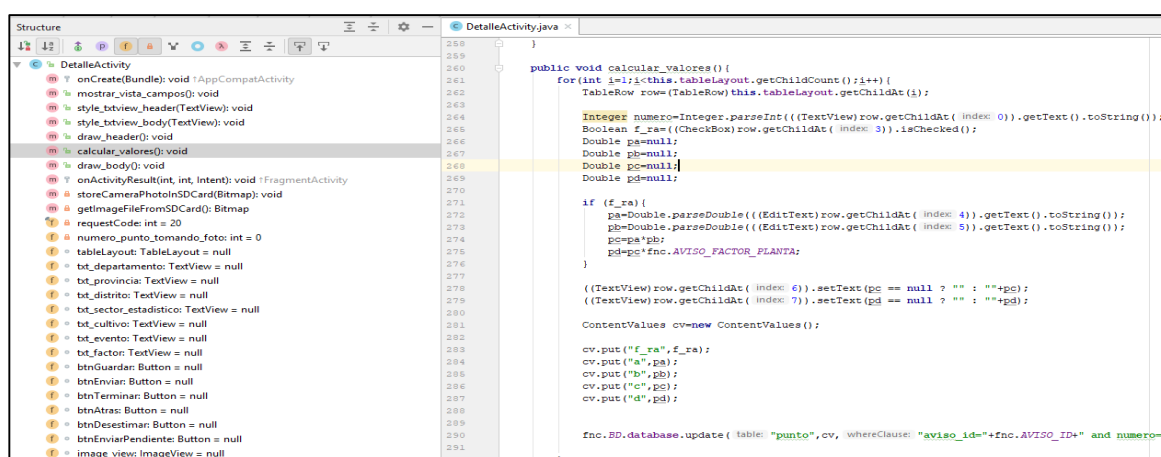


Imagen 57: Codificación del proceso principal para registrar datos de siniestro por aviso en la Tablet de la HUN°7
Fuente: Creación propia

ii. Resultado:

En la imagen 58 mostramos el resultado de la codificación del proceso principal para registrar datos de siniestro por aviso en la Tablet

TesisApp						
ATRÁS		GUARDAR		DESESTIMAR		TERMINAR
Departamento	CAJAMARCA	Provincia	CAJABAMBA	Distrito	CAJABAMBA	Sector
Evento	INUNDACIÓN					
N°	Este	Norte	RqA	N° Plantas	Cultivos por Planta	Cultivos p
1	-78°6'52.7716"	-7°37'22.1810"	✓	3	2	6
2	-78°6'52.7716"	-7°36'47.1208"	□			
3	-78°5'51.2170"	-7°37'10.0322"	□			
4	-78°5'51.2170"	-7°37'10.0322"	□			

Imagen 58: Resultado de la Codificación de la HU N°7
Fuente: Creación propia

b. HU N° 8: ENVIAR AVISOS TERMINADOS – TABLET

i. Proceso:

En la imagen 59 mostramos la Codificación del proceso principal para enviar avisos terminados de la Tablet al servidor.

```

419 public static void enviar_avisos(final Long id){
420     if (fnc.get_estado_avisos(id).equals("terminado")){
421         final Long sid=fnc.get_sid_avisos(id);
422         Cursor cursor = fnc.BD.database.query( table: "punto", new String[]{"numero", "f_ra", "a", "b"},
423         String data_json="";
424         while (cursor.moveToNext()){
425             boolean f_ra=(cursor.getInt( columnIndex: 1)==1);
426             if (f_ra) {
427                 data_json += "{\n\n": " + cursor.getInt( columnIndex: 0) + ",\nf\n": "+(f_ra?1:0)+"\n\n";
428             }
429         }
430         cursor.close();
431         System.out.println(data_json);
432
433         final String data_json_enviar="["+data_json.substring(0,data_json.length()-1)+"]";
434
435         AsyncTask asyncTask = (AsyncTask) (params) -> {
436             try {
437                 String data = URLEncoder.encode( s: "d", enc: "UTF-8") + "=" + URLEncoder.encode( s: "i", enc: "UTF-8") + "=" + URLEncoder.encode( s: "f_ra", enc: "UTF-8") + "=" + URLEncoder.encode( s: "a", enc: "UTF-8") + "=" + URLEncoder.encode( s: "b", enc: "UTF-8");
438                 data = data + "&" + URLEncoder.encode( s: "i", enc: "UTF-8") + "=" + URLEncoder.encode( s: "f_ra", enc: "UTF-8") + "=" + URLEncoder.encode( s: "a", enc: "UTF-8") + "=" + URLEncoder.encode( s: "b", enc: "UTF-8");
439
440                 URL url = new URL( spec: HOST + "ajax/cargar_avisos/");
441                 URLConnection conn = url.openConnection();
442                 conn.setDoOutput(true);
443                 OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());
444             }
445         }
446     }

```

Imagen 59: Codificación del proceso principal para enviar avisos terminados de la Tablet al servidor de la HUN°8

Fuente: Creación propia

ii. Resultado:

En la imagen 60 mostramos el resultado de la Codificación del proceso principal para enviar avisos terminados de la Tablet al servidor.

Evento		Estado	Factor Planta	Dese
IANA	SEQUÍA	pendiente	0.123	No
	DESLIZAMIENTO	pendiente	0.05	No
RRAL	TEMPERATURAS EXTREMAS	pendiente	0.023	No
BO	HUAYCO	pendiente	0.526	No
BA	INUNDACIÓN	enviado	0.01	No

Imagen 60: Resultado de la Codificación de la HU N°8

Fuente: Creación propia

c. HU N° 9: AVISOS ENVIADOS

i. Proceso:

En la imagen 61 mostramos la codificación del proceso principal para visualizar la bandeja de avisos enviados desde la Tablet.

```
151 public void ver_data(){
152     String ver_estado = spinner.getSelectedItem().toString().toLowerCase();
153     String filtro=this.txt_filtro.getText().toString();
154     Cursor cursor=null;
155     String mas_filtros="";
156     mas_filtros+=" and (departamento like '%"+filtro+"%' or provincia like '%"+filtro+"%' or distrito like '%"+
157     if (!ver_estado.equals("todos")){
158         mas_filtros+=" and estado='"+ver_estado+"';
159     }
160     cursor = fnc.BD.database.query( table: "aviso", new String[]{"sid", "cultivo", "departamento", "provincia", "dist"
```

Imagen 61: Codificación del proceso principal para visualizar la bandeja de avisos enviados desde la Tablet de la HUN°9

Fuente: Creación propia

ii. Resultado:

En la imagen 62 mostramos el resultado de la codificación del proceso principal para visualizar la bandeja de avisos enviados desde la Tablet.

TesisApp								
DESCARGAR DATOS CAMAYA		Enviado ▼		Buscar...		BUSCAR		
N°	Codigo	Cultivo	Departamento	Provincia	Distrito	Sector	Evento	Estado
1	21	PAPA	CAJAMARCA	CAJABAMBA	CAJABAMBA	CAJABAMBA	INUNDACIÓN	enviado

Imagen 62: Resultado de la Codificación de la HU N°9

Fuente: Creación propia

d. HU N°10: CERRAR EL AVISO – WEB

i. Proceso:

En la imagen 63 mostramos la codificación del proceso principal para cerrar aviso desde la web.

```
312         if request.POST.has_key("btnCerrarEvaluacion"):
313             aviso.f_evaluador=False
314             aviso.f_fin=True
315             aviso.save()
316
```

Imagen 63: Codificación del proceso principal para cerrar aviso desde la web de la HUN°10
Fuente: Creación propia

ii. Resultado:

En la imagen 64 mostramos el resultado de la codificación del proceso principal para cerrar aviso desde la web.

Detalle de Aviso

Detalle

Puntos

Departamento	CAJAMARCA	Provincia	CAJAMARCA
Distrito	CAJAMARCA	Sector	AGOCUCHO
Cultivo	MAIZ AMILACIO	Tipo Evento	INUNDACIÓN
Estado	Finalizo		
Factor	5.0000	Pago por Factor	90.0000
Minimo Siembra	100.0000	Total Siembra	2385.0000
Evaluador	Cinthy Amaya (camaya)		
Fecha Evaluación	14/03/2018		
Resultado	No Indemnizable Monto Pagar —		

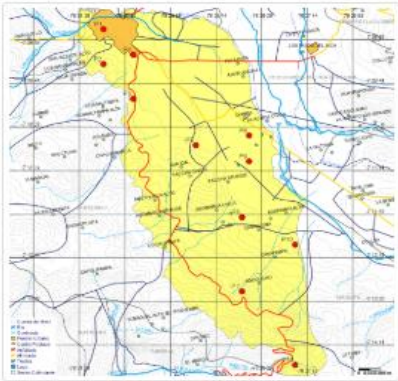


Imagen 64: Resultado de la Codificación de la HU N°10
Fuente: Creación propia

D. ITERACIÓN IV

a. HU N° 11: ENVIAR AVISO A POSITIVA

i. Proceso:

En la imagen 65 mostramos la codificación del proceso principal para enviar avisos a positiva.

```
255
256
257
258
259
if request.POST.has_key("btnRetornar"):
    aviso.f_ajustador=False
    aviso.f_positiva=True
    aviso.save()
```

Imagen 65: Codificación del proceso principal para enviar avisos a positiva de la HUN° 11
Fuente: Creación propia

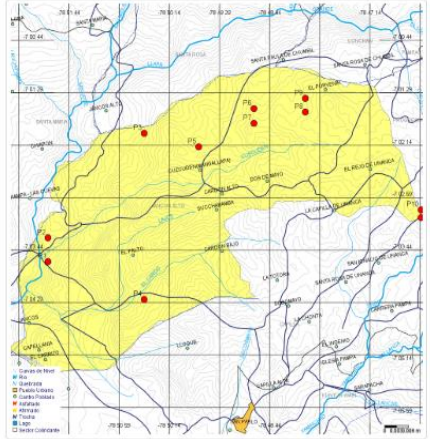
ii. Resultado:

En la imagen 66 mostramos el resultado de la codificación del proceso principal para enviar avisos a positiva.

Detalle de Aviso

Detalle Puntos

Departamento	CAJAMARCA	Provincia	SAN PABLO
Distrito	SAN PABLO	Sector	JANCOS ALTO
Cultivo	MAIZ AMILACIO	Tipo Evento	TEMPERATURAS EXTREMAS
Estado	Miembro		
Factor	—	Pago por Factor	—
Minimo Siembra	—	Total Siembra	—
Evaluador	— ()		
Fecha Evaluación	—		



Retornar a Responsable de Región Guardar Enviar a Positiva

Imagen 66: Resultado de la Codificación de la HU N° 11
Fuente: Creación propia

b. HU N° 12: ENVIAR AVISO AL AJUSTADOR

i. Proceso:

En la imagen 67 mostramos la codificación del proceso principal para enviar los avisos al ajustador

```
233         if request.POST.has_key("btnEnviar"):
234             aviso.f_positiva=False
235             aviso.f_ajustador=True
236             aviso.save()
237
```

Imagen 67: Codificación del proceso principal para enviar los avisos al ajustador de la HUN°12
Fuente: Creación propia

ii. Resultado:

En la imagen 68 mostramos el resultado de la codificación del proceso principal para enviar los avisos al ajustador.

Detalle		Puntos	
Departamento	CAJAMARCA	Provincia	CUTERVO
Distrito	SAN LUIS DE LUCMA	Sector	SAIREPAMPA
Cultivo	PAPA	Tipo Evento	HUAYCO
Estado	Positivo		
Factor	0.50	Pago por Factor	500
Mínimo Siembra	200	Total Siembra	—
Evaluador	— ()		
Fecha Evaluación	—		

[Retornar a Minagri](#)
[Generar Puntos](#)
[Guardar](#)
[Enviar a Ajustador](#)

Imagen 68:Resultado de la Codificación de la HU N°12
Fuente: Creación propia

c. HU N° 13: ENVIAR AVISO AL EVALUADOR

i. Proceso:

En la imagen 69 mostramos la Codificación del proceso principal para enviar los avisos al evaluador

```
260         if request.POST.has_key("btnEnviar"):
261             aviso.f_ajustador=False
262             aviso.f_evaluador=True
263             aviso.save()
```

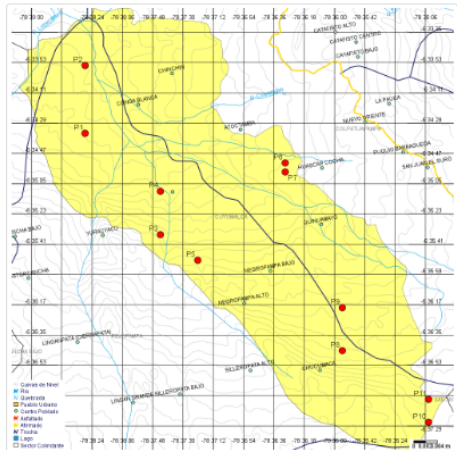
Imagen 69: Codificación del proceso principal para enviar los avisos al evaluador de la HUN°13
Fuente: Creación propia

ii. Resultado:

En la imagen 70 mostramos el resultado de la Codificación del proceso principal para enviar los avisos al evaluador.

Detalle de Aviso

Detalle				Puntos	
Departamento	CAJAMARCA	Provincia	CHOTA		
Distrito	CHOTA	Sector	CUYUMALCA		
Cultivo	PAPA	Tipo Evento	SEQUIA		
Estado	<button>Ajustador</button>				
Factor	0.0500	Pago por Factor	500.0000		
Mínimo Siembra	100.0000	Total Siembra	—		
Evaluador	<div>Cinthy Amaya (camaya)</div> <div>—</div> <div>Cinthy Amaya (camaya)</div> <div>Amador Rodas (acrodas)</div>				
Fecha Evaluación					



Retornar a Positiva Guardar Enviar a Evaluador

Imagen 70: Resultado de la Codificación de la HU N°13
Fuente: Creación propia

E. ITERACIÓN IV

a. HU N°14: RETORNAR AVISOS AL AJUSTADOR

i. Proceso:

En la imagen 71 mostramos la codificación del proceso principal para retornar los avisos al ajustador.

```
308         if request.POST.has_key("btnRetornar"):
309             aviso.f_evaluador=False
310             aviso.f_ajustador=True
311             aviso.save()
```

Imagen 71: Codificación del proceso principal para retornar los avisos al ajustador de la HUN°14
Fuente: Creación propia

ii. Resultado:

En la imagen 72 mostramos el resultado de la codificación del proceso principal para retornar los avisos al ajustador.

Detalle			
Puntos			
Departamento	CAJAMARCA	Provincia	JAEN
Distrito	SALLIQUE	Sector	CHALANMACHE
Cultivo	MAIZ AMILACIO	Tipo Evento	GRANIZOS
Estado	Evaluador		
Factor	0.2560	Pago por Factor	200.0000
Minimo Siembra	10000.0000	Total Siembra	
Evaluador	-- ()		
Fecha Evaluación	30/04/2018		

Retornar a Ajustador | Guardar | Desestimar | Cerrar Evaluacion

Imagen 72: Resultado de la Codificación de la HU N°14
Fuente: Creación propia

b. HU N°15: RETORNAR AVISOS A POSITIVA.

i. Proceso:

En la imagen 73 mostramos la codificación del proceso principal para retornar los avisos a positiva.

```
255
256         if request.POST.has_key("btnRetornar"):
257             aviso.f_ajustador=False
258             aviso.f_positiva=True
259             aviso.save()
```

Imagen 73: Codificación del proceso principal para retornar los avisos a positiva de la HUN°15
Fuente: Creación propia

ii. Resultado:

En la imagen 74 mostramos el resultado de la codificación del proceso principal para retornar los avisos a positiva.

The screenshot displays a web application interface with a form on the left and a map on the right. The form is titled 'Detalle' and 'Puntos'. It contains the following fields:

Detalle		Puntos	
Departamento	CAJAMARCA	Provincia	CHOTA
Distrito	CHOTA	Sector	CUYUMALCA
Cultivo	PAPA	Tipo Evento	SEQUÍA
Estado	<button>Ajustador</button>		
Factor	0.0500	Pago por Factor	500.0000
Minimo Siembra	100.0000	Total Siembra	—
Evaluador	Cinthya Amaya (camaya)		
Fecha Evaluación	20/07/2019		

At the bottom of the form, there are three buttons: 'Retornar a Positiva' (blue), 'Guardar' (green), and 'Enviar a Evaluador' (red).

The map on the right shows a geographical area with a grid overlay. It includes labels for 'Cajamarca', 'Chota', and 'Cuyumalca'. The map is color-coded with yellow and green areas, and red dots are placed at various locations. A legend at the bottom left of the map identifies symbols for 'Cajamarca', 'Chota', 'Cuyumalca', 'Lago', and 'Tronco'.

Imagen 74: Resultado de la Codificación de la HU N°15
Fuente: Creación propia

c. HU N° 16: RETORNAR AVISOS A MINAGRI

i. Proceso:

En la imagen 75 mostramos la codificación del proceso principal para retornar los avisos a MINAGRI.

```
228
229
230
231
232
if request.POST.has_key("btnRetornar"):
    aviso.f_positiva=False
    aviso.f_minagri=True
    aviso.save()
```

Imagen 75: Codificación del proceso principal para retornar los avisos a MINAGRI de la HU N° 16
Fuente: Creación propia

ii. Resultado:

En la imagen 76 mostramos el resultado de la codificación del proceso principal para retornar los avisos a MINAGRI.

The screenshot displays a web application interface with a form on the left and a map on the right. The form is titled 'Detalle' and 'Puntos'. It contains the following fields:

Detalle		Puntos	
Departamento	CAJAMARCA	Provincia	CUTERVO
Distrito	SAN LUIS DE LUCMA	Sector	SAIREPAMPA
Cultivo	PAPA	Tipo Evento	HUAYCO
Estado	Positiva		
Factor	<input type="text"/>	Pago por Factor	<input type="text"/>
Minimo Siembra	<input type="text"/>	Total Siembra	—
Evaluador	— ()		
Fecha Evaluación	—		

At the bottom of the form, there are four buttons: 'Retornar a Minagri' (blue), 'Generar Puntos' (green), 'Guardar' (green), and 'Enviar a Ajustador' (red).

The map on the right shows a geographical area with a grid overlay. A yellow shaded region indicates the area of interest. Red dots are placed on the map, likely representing the locations of the events. The map includes labels for various locations and features.

Imagen 76: Resultado de la Codificación de la HU N° 16
Fuente: Creación propia

d. HU N° 17: RETORNAR AVISOS AL REPRESENTANTE DE REGIÓN

i. Proceso:

En la imagen 77 mostramos la codificación del proceso principal para retornar los avisos a al representante de región.

```
191         if request.POST.has_key("btnRetornar"):  
192             aviso.f_minagri=False  
193             aviso.f_responsable_region=True  
194             aviso.save()
```

Imagen 77: Codificación del proceso principal para retornar los avisos a al representante de región de la HUN°17

Fuente: Creación propia

ii. Resultado:

En la imagen 78 mostramos la codificación del proceso principal para retornar los avisos a al representante de región.

The screenshot displays a web application interface with a form on the left and a map on the right. The form is titled 'Detalle' and 'Puntos'. It contains the following fields:

Departamento	Provincia
CAJAMARCA	SAN PABLO

Distrito	Sector
SAN PABLO	JANCOS ALTO

Cultivo	Tipo Evento
MAIZ AMILACIO	TEMPERATURAS EXTREMAS

Estado: Minagri

Factor: — Pago por Factor: —

Minimo Siembra: — Total Siembra: —

Evaluador: — ()

Fecha Evaluación: —

At the bottom of the form, there are three buttons: 'Retornar a Responsable de Región', 'Guardar', and 'Enviar a Positiva'.

The map on the right shows a geographical area with a yellow highlighted region, likely representing the area of interest for the event.

Imagen 78: Resultado de la Codificación de la HU N°17

Fuente: Creación propia

F. ITERACIÓN V

a. HU N° 18: ASIGNAR EVALUADOR

i. Proceso:

En la imagen 79 mostramos la codificación del proceso principal para asignar evaluador.

```
248
249
250         if request.POST.has_key("evaluador"):
251             if evaluadores.filter(id=request.POST["evaluador"]).exists():
252                 aviso.evaluador_id=request.POST["evaluador"]
253             else:
254                 aviso.evaluador=None
255             aviso.save()
```

Imagen 79: Codificación del proceso principal para asignar evaluador de la HUN°18
Fuente: Creación propia

ii. Resultado:

En la imagen 80 mostramos el resultado de la codificación del proceso principal para asignar evaluador.

The screenshot displays the 'Detalle de Aviso' (Notice Detail) form. It includes a table with fields for Departamento (CAJAMARCA), Provincia (CHOTA), Distrito (CHOTA), Sector (CUYUMALCA), Cultivo (PAPA), Tipo Evento (SEQUIA), Estado (Ajustador), Factor (0.0500), Pago por Factor (500.0000), Minimo Siembra (100.0000), and Total Siembra (—). The Evaluador field is a dropdown menu with 'Cinthy Amaya (camaya)' selected. The Fecha Evaluación field is empty. To the right of the form is a map of the region. At the bottom right, there are three buttons: 'Retornar a Positiva', 'Guardar', and 'Enviar a Evaluador'.

Detalle de Aviso			
Detalle		Puntos	
Departamento	CAJAMARCA	Provincia	CHOTA
Distrito	CHOTA	Sector	CUYUMALCA
Cultivo	PAPA	Tipo Evento	SEQUIA
Estado	Ajustador		
Factor	0.0500	Pago por Factor	500.0000
Minimo Siembra	100.0000	Total Siembra	—
Evaluador	Cinthy Amaya (camaya)		
Fecha Evaluación			

Imagen 80: Resultado de la Codificación de la HU N°18
Fuente: Creación propia

b. HU N° 19: PROGRAMAR EVALUACIÓN

i. Proceso:

En la imagen 81 mostramos la codificación del proceso principal para programar evaluación.

```
240     if request.POST.has_key("btnEnviar") or request.POST.has_key("btnGuardar"):
241         if request.POST.has_key("fecha_evaluacion"):
242             try:
243                 fecha_evaluacion=datetime.datetime.strptime(request.POST["fecha_evaluacion"], '%d/%m/%Y').date()
244             except Exception as e:
245                 fecha_evaluacion=None
246                 aviso.fecha_evaluacion=fecha_evaluacion
247                 aviso.save()
```

Imagen 81: Codificación del proceso principal para programar evaluación de la HUN°19
Fuente: Creación propia

ii. Resultado:

En la imagen 82 mostramos el resultado de la codificación del proceso principal para programar evaluación.

Departamento	CAJAMARCA	Provincia	SAN MIGUEL
Distrito	SAN GREGORIO	Sector	SAN GREGORIO
Cultivo	MAIZ AMILACIO	Tipo Evento	INUNDACIÓN
Estado			
Factor	500.0000		
Minimo Siembra	—		
Evaluador			
Fecha Evaluación	14/07/2019		

< Julio 2019 >

Do	Lu	Ma	Mi	Ju	Vi	Sa
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Imagen 82: Resultado de la Codificación de la HU N°19
Fuente: Creación propia

c. HU N° 20: AGREGAR DATOS A AVISO

i. Proceso:

En la imagen 83 mostramos la codificación del proceso principal para agregar datos.

```
205 if request.POST.has_key("factor_planta") and request.POST.has_key("pago_por_factor") and request.POST.has_key("minimo_siembra"):
206     try:
207         aviso.factor_planta=(float)(request.POST["factor_planta"])
208     except Exception as e:
209         aviso.factor_planta=None
210     try:
211         aviso.pago_por_factor=(float)(request.POST["pago_por_factor"])
212     except Exception as e:
213         aviso.pago_por_factor=None
214     try:
215         aviso.minimo_siembra=(float)(request.POST["minimo_siembra"])
216     except Exception as e:
217         aviso.minimo_siembra=None
218     aviso.save()
219
```

Imagen 83:Codificación del proceso principal para agregar datos de la HUN°20
Fuente: Creación propia

ii. Resultado:

En la imagen 84 mostramos el resultado de la codificación del proceso principal para agregar datos.

Departamento	CAJAMARCA	Provincia	CUTERVO
Distrito	SAN LUIS DE LUCMA	Sector	SAIREPAMPA
Cultivo	PAPA	Tipo Evento	HUAYCO
Estado	Positivo		
Factor	0.01	Pago por Factor	500
Minimo Siembra	200	Total Siembra	—
Evaluador	— ()		
Fecha Evaluación	—		

Imagen 84:Resultado de la Codificación de la HU N°20
Fuente: Creación propia

d. HU N° 21: EVALUAR AVISOS VÍA WEB

i. Proceso:

En la imagen 85 mostramos la codificación del del proceso principal para evaluar avisos vía web.

```
268 total_siembra=0
269 for punto in puntos:
270     if request.POST.has_key("f_registra_analisis_%s" % punto.numero):
271         f_registra_analisis=request.POST["f_registra_analisis_%s" % punto.numero]
272         if f_registra_analisis=="on":
273             if request.POST.has_key("numero_plantas_%s" % punto.numero) or request.POST.has_key("cultivo_plantas_%s" % punto.numero):
274                 try:
275                     numero_plantas=Decimal(request.POST["numero_plantas_%s" % punto.numero])
276                 except Exception as e:
277                     numero_plantas=None
278                 try:
279                     cultivo_plantas=Decimal(request.POST["cultivo_plantas_%s" % punto.numero])
280                 except Exception as e:
281                     cultivo_plantas=None
282
283                 if numero_plantas!=None and cultivo_plantas!=None:
284                     a=numero_plantas
285                     b=cultivo_plantas
286                     c=a*b
287                     d=aviso.factor_planta*c
288
289                     punto.f_registra_analisis=True
290                     punto.numero_plantas=a
291                     punto.cultivo_plantas=b
292                     punto.calculo=c
293                     punto.siembra=d
294                     punto.save()
295
296                     total_siembra+=d
297
298 if aviso.minimo_siembra<total_siembra:
299     aviso.f_indemnizable=False
300     aviso.monto_pagar=None
301 else:
302     aviso.f_indemnizable=True
303     aviso.monto_pagar=(aviso.minimo_siembra-total_siembra)*aviso.pago_por_factor
304
305 aviso.total_siembra=total_siembra
306 aviso.save()
```

Imagen 85:ificación del proceso principal para evaluar avisos vía web de la HUN°21

Fuente: Creación propia

ii. Resultado:

En la imagen 86 mostramos el resultado de la codificación del del proceso principal para evaluar avisos vía web.

Análisis	Posición Este	Posición Norte	Registra Análisis	N° Plantas (<100)	N° Cultivos por Planta (<25)	N° Cultivos por análisis	Siembra
1	-79°14'33.8596"	-5°36'37.6726"	<input checked="" type="checkbox"/>	54	5	270	33.21
2	-79°14'33.8596"	-5°35'56.4300"	<input checked="" type="checkbox"/>	7	8	56	6.888
3	-79°12'13.8745"	-5°39'2.7225"	<input checked="" type="checkbox"/>	5	7	35	4.305
4	-79°12'13.8745"	-5°41'35.1235"	<input checked="" type="checkbox"/>	8	9	72	8.856
5	-79°8'51.9269"	-5°40'8.1461"	<input type="checkbox"/>				
6	-79°7'9.5367"	-5°38'45.8671"	<input type="checkbox"/>				
7	-79°7'9.5367"	-5°40'50.2199"	<input type="checkbox"/>				
8	-79°6'17.6159"	-5°39'20.0340"	<input type="checkbox"/>				
9	-79°6'17.6159"	-5°43'17.8416"	<input type="checkbox"/>				
10	-79°3'28.4682"	-5°42'51.1224"	<input type="checkbox"/>				
11	-79°3'28.4682"	-5°42'58.1969"	<input type="checkbox"/>				
Total							53.259

Imagen 86:Resultado de la Codificación de la HU N°21

Fuente: Creación propia

G. ITERACIÓN VII

a. HU N° 22: INICIO DE SESIÓN

i. Proceso:

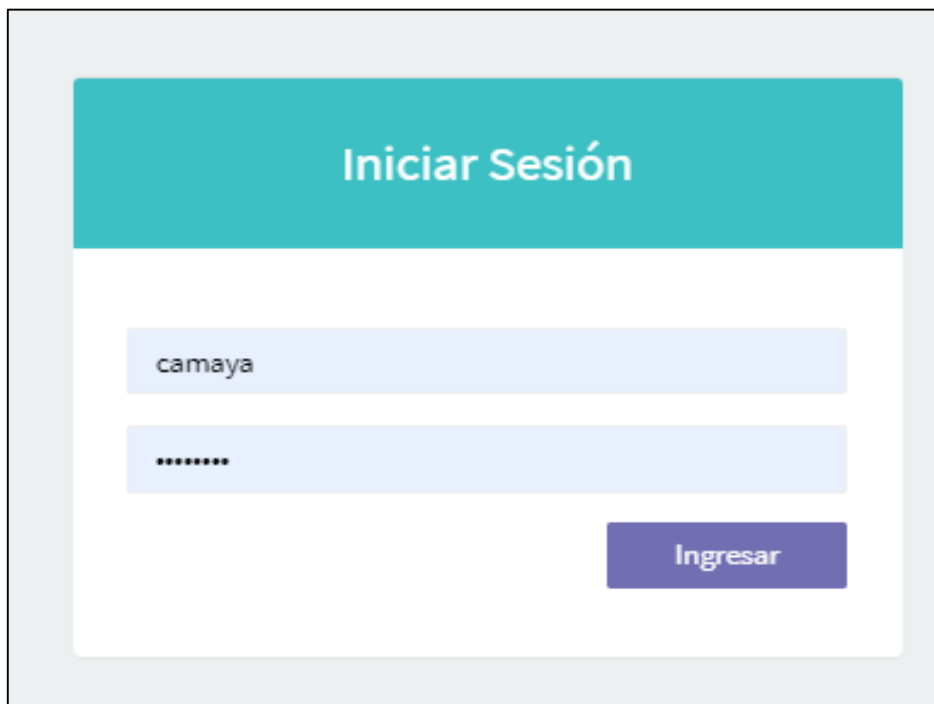
En la imagen 87 mostramos la codificación del proceso principal para el inicio de sesión.

```
39 usuario=request.POST['usuario']
40 clave=request.POST['clave']
41 user = auth.authenticate(username=usuario, password=clave)
42 loglogin=LogLogin.objects.create(remote_addr=fnc.utils.get_remote_addr_from_request(request),usuario_texto=usuario)
43 if user is not None:
44     loglogin.usuario=user
45     loglogin.save()
46     if user.is_active:
47         auth.login(request, user)
48         return fnc.redireccionar.home()
49     else:
50         mensaje='Usuario no se encuentra activo.'
51 else:
52     mensaje='Usuario o clave no son validos.'
53 else:
54     mensaje='Usuario o clave incorrecta.'
```

Imagen 87:Codificación del proceso principal para el inicio de sesión de la HUN°22
Fuente: Creación propia

ii. Resultado:

En la imagen 88 mostramos el resultado de la codificación del proceso principal para el inicio de sesión.



The image shows a web interface for logging in. At the top, there is a teal header with the text "Iniciar Sesión". Below this, there is a white form area. Inside the form, there are two light blue input fields. The first field contains the text "camaya". The second field contains seven dots, indicating a password. Below these fields, there is a purple button with the text "Ingresar".

Imagen 88:Resultado de la Codificación de la HU N°22
Fuente: Creación propia

b. HU N° 23: BUSCAR AVISO

i. Proceso:

En la imagen 89 mostramos la codificación del proceso principal para buscar aviso.

```
135
136     if request.GET and request.GET.has_key("estado_filtro"):
137         estado_filtro=request.GET["estado_filtro"].lower()
138         if not estado_filtro in ["responsable_region","minagri","positiva","ajustador","evaluador","fin"]:
139             estado_filtro="todos"
140     avisos=Aviso.objects.all()
141     if estado_filtro!="todos":
142         if estado_filtro=="responsable_region":
143             avisos=avisos.filter(f_responsable_region=True)
144         elif estado_filtro=="minagri":
145             avisos=avisos.filter(f_minagri=True)
146         elif estado_filtro=="positiva":
147             avisos=avisos.filter(f_positiva=True)
148         elif estado_filtro=="ajustador":
149             avisos=avisos.filter(f_ajustador=True)
150         elif estado_filtro=="evaluador":
151             avisos=avisos.filter(f_evaluador=True)
152         elif estado_filtro=="fin":
153             avisos=avisos.filter(f_fin=True)
154
```

Imagen 89: Codificación del proceso principal para buscar aviso de la HU N°23
Fuente: Creación propia

ii. Resultado:

En la imagen 90 mostramos el resultado de la codificación del proceso principal para buscar aviso.

Listar Aviso (1)									
Minagri		jan		Buscar					
Cultivo	Región	Distrito	Provincia	Sector	Evento	Creado	Resultado	Estado	Detalle
MAIZ AMILACIO	CAJAMARCA	SAN PABLO	SAN PABLO	JANCOS ALTO	TEMPERATURAS EXTREMAS	Mar. 13, 2018, 12:53 p.m.	En Proceso	Minagri	Ver Detalle

Imagen 90: Resultado de la Codificación de la HU N°23
Fuente: Creación propia

c. HU N° 24: EXPORTAR AVISOS EN EXCEL

i. Proceso:

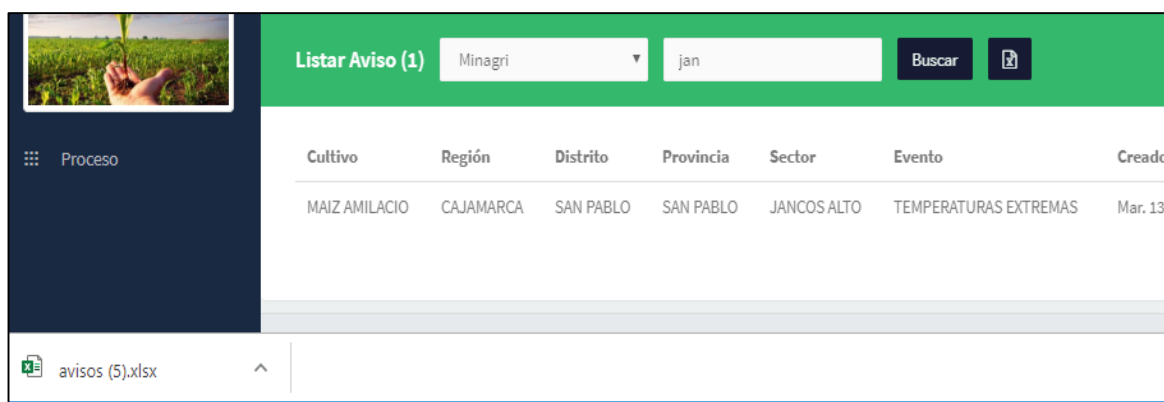
En la imagen 91 mostramos la codificación del proceso principal para exportar avisos en Excel.

```
337     data=[
338         ("#", "codigo", "cultivo", "departamento", "provincia", "distrito", "sector estadístico", "tipo evento", "factor", "desestimado", "creado", "evaluado", "indemnizable", "estado")
339     ]
340     c=0
341     for a in Aviso.objects.all():
342         c+=1
343         data.append(
344             (c,a.id,a.cultivo,a.sector_estadístico.distrito.provincia.departamento.nombre,a.sector_estadístico.distrito.provincia.nombre,a.sector_estadístico.distrito.nombre,a.sector_estadístico.tipo_evento,a.sector_estadístico.factor,a.sector_estadístico.desestimado,a.sector_estadístico.creado,a.sector_estadístico.evaluado,a.sector_estadístico.indemnizable,a.sector_estadístico.estado)
345         )
346     return ExcelResponse(data,"avisos")
```

Imagen 91:Codificación del proceso principal para exportar avisos en Excel de la HUN°24
Fuente: Creación propia

ii. Resultado:

En la imagen 92 mostramos el resultado de la codificación del proceso principal para exportar avisos en Excel.



The screenshot shows a web application interface. On the left is a sidebar with a menu icon and the text 'Proceso'. The main area has a green header with the text 'Listar Aviso (1)' and a dropdown menu showing 'Minagri'. Below the header is a table with the following columns: Cultivo, Región, Distrito, Provincia, Sector, Evento, and Creado. The table contains one row of data: MAIZ AMILACIO, CAJAMARCA, SAN PABLO, SAN PABLO, JANCOS ALTO, TEMPERATURAS EXTREMAS, and Mar. 13, 2023. To the right of the table is a 'Buscar' button and a download icon. At the bottom of the interface, there is a file download bar showing 'avisos (5).xlsx' with an upward arrow icon.

Cultivo	Región	Distrito	Provincia	Sector	Evento	Creado
MAIZ AMILACIO	CAJAMARCA	SAN PABLO	SAN PABLO	JANCOS ALTO	TEMPERATURAS EXTREMAS	Mar. 13, 2023

Imagen 92:Resultado de la Codificación de la HU N°24
Fuente: Creación propia

H. ITERACIÓN VIII

a. HU N° 25: DESCARGAR ACTAS DE ATENCIÓN POR AVISO

i. Proceso:

En la imagen 93 mostramos la codificación del proceso principal para descargar actas de atención por aviso

```
99 def get_response(aviso):
100     response = HttpResponse(content_type='application/pdf')
101     response['Content-Disposition'] = 'attachment; filename=aviso_%s.pdf' % (aviso.id,)
102     canva=Canvas.Canvas(response)
103
104     draw_header(canva,aviso)
105     draw_body(canva,aviso)
106     draw_puntos(canva,aviso)
107     draw_footer(canva,aviso)
108
109     canva.save()
110     return response
```

Imagen 93:Codificación del proceso principal para descargar actas de atención por aviso de la HUN°25
Fuente: Creación propia

ii. Resultado:

En la imagen 94 mostramos el resultado de la codificación del proceso principal para descargar actas de atención por aviso.

FECHA DE EVALUACIÓN: 2018-03-14

NOTA: Con la firma anterior se hace constar la visita del inspector(pj) para la atención del aviso de siniestro, en el marco del Seguro Agrario.

DATOS GENERALES

CÓDIGO: 15 CULTIVO: MAÍZ AMILADO
REGIÓN: CALAMARCA PROVINCIA: CALAMARCA DISTRITO: CALAMARCA SECTOR ESTADÍSTICO: AGROPECUARIO

EVALUACIÓN

ANÁLISIS	1	2	3	4	5	6	7	8	9	10	11
Registro Análisis	SI	SI	SI	SI	NO	SI	NO	SI	NO	SI	NO

Factor: 5.0000

Punto	Este	Norte	Registro Análisis	N° Plantas	N° Cultivos por Planta	N° Cultivos por punto	Siembras
1	-78°31'5.0750"	-7°54'5.9750"	SI	6.0000	10.0000	60.0000	300.0000
2	-78°31'5.0750"	-7°52'4.9087"	SI	2.0000	6.0000	12.0000	60.0000
3	-78°30'36.7652"	-7°10'0.8163"	SI	5.0000	15.0000	75.0000	375.0000
4	-78°30'36.7652"	-7°51'5.3273"	SI	6.0000	18.0000	108.0000	540.0000
5	-78°29'34.9490"	-7°10'48.7349"	NO	—	—	—	—
6	-78°28'49.3113"	-7°12'3.0717"	SI	7.0000	21.0000	147.0000	735.0000
7	-78°28'49.3113"	-7°13'19.6421"	NO	—	—	—	—
8	-78°28'42.3460"	-7°10'38.8509"	SI	4.0000	12.0000	48.0000	240.0000
9	-78°28'42.3460"	-7°11'5.3607"	NO	—	—	—	—
10	-78°27'56.8708"	-7°12'31.4257"	SI	3.0000	9.0000	27.0000	135.0000
11	-78°27'56.8708"	-7°14'35.6420"	NO	—	—	—	—
Total Siembras							2385.0000

NO INDEMNIZABLE

Fecha de evaluación : 2018-03-14

Evaluador : camaya Firma de representante

Imagen 94:Resultado de la Codificación de la HU N°25
Fuente: Creación propia

b. HU N° 26: VER ESTADO DE AVISOS

i. Proceso:

En la imagen 95 mostramos la codificación del proceso principal para ver estados de los avisos.

```
62 if fnc.session.valida(request):
63     son_en_proceso=Aviso.objects.filter(f_fin=False).count()
64     son_terminado_indemnizable=Aviso.objects.filter(f_indemnizable=True,f_fin=True,f_desestimado=False).count()
65     son_terminado_no_indemnizable=Aviso.objects.filter(f_indemnizable=False,f_fin=True,f_desestimado=False).count()
66     son_terminado_desestimado=Aviso.objects.filter(f_fin=True,f_desestimado=True).count()
67
68
69     son_f_responsable_region=Aviso.objects.filter(f_responsable_region=True).count()
70     son_f_minagri=Aviso.objects.filter(f_minagri=True).count()
71     son_f_positiva=Aviso.objects.filter(f_positiva=True).count()
72     son_f_ajustador=Aviso.objects.filter(f_ajustador=True).count()
73     son_f_evaluador=Aviso.objects.filter(f_evaluador=True).count()
74     son_f_fin=Aviso.objects.filter(f_fin=True).count()
75     return fnc.render.run(request,"main/home.html",{
76         "son_en_proceso":son_en_proceso,
77         "son_terminado_indemnizable":son_terminado_indemnizable,
78         "son_terminado_no_indemnizable":son_terminado_no_indemnizable,
79         "son_terminado_desestimado":son_terminado_desestimado,
80
81         "son_f_responsable_region":son_f_responsable_region,
82         "son_f_minagri":son_f_minagri,
83         "son_f_positiva":son_f_positiva,
84         "son_f_ajustador":son_f_ajustador,
85         "son_f_evaluador":son_f_evaluador,
86         "son_f_fin":son_f_fin,
87     })
88
89 return fnc.redirect("login")
```

Imagen 95:Codificación del proceso principal para ver estados de los avisos de la HUN°26

Fuente: Creación propia

ii. Resultado:

En la imagen 96 mostramos el resultado de la codificación del proceso principal para ver estados de los avisos. Todos los usuarios verán la misma interfaz, pero sólo podrán ingresar según permisos.



Imagen 96:Resultado de la Codificación de la HU N°26

Fuente: Creación propia

c. HU N° 27: LISTAR AVISOS

i. Proceso:

En la imagen 97 mostramos la codificación del proceso principal para listar los avisos.

```

127
128 def listar_avisos(request):
129     if fnc.session.valida(request):
130         txt_filtro=None
131         estado_filtro="todos"
132         if request.GET and request.GET.has_key("txt_filtro"):
133             txt_filtro=request.GET["txt_filtro"].strip()
134
135
136         if request.GET and request.GET.has_key("estado_filtro"):
137             estado_filtro=request.GET["estado_filtro"].lower()
138             if not estado_filtro in ["responsable_region", "minagri", "positiva", "ajustador", "evaluador", "fin"]:
139                 estado_filtro="todos"
140
141     avisos=Aviso.objects.all()
142     if estado_filtro=="todos":
143         if estado_filtro=="responsable_region":
144             avisos=avisos.filter(f_responsable_region=True)
145         elif estado_filtro=="minagri":
146             avisos=avisos.filter(f_minagri=True)
147         elif estado_filtro=="positiva":
148             avisos=avisos.filter(f_positiva=True)
149         elif estado_filtro=="ajustador":
150             avisos=avisos.filter(f_ajustador=True)
151         elif estado_filtro=="evaluador":
152             avisos=avisos.filter(f_evaluador=True)
153         elif estado_filtro=="fin":
154             avisos=avisos.filter(f_fin=True)
155
156     if txt_filtro!=None:
157         avisos=avisos.filter(Q(cultivo__icontains=txt_filtro) | Q(tipo_evento__icontains=txt_filtro) | Q(sector_estadistico_nombre__icontains=txt_filtro))
158
159     return fnc.render.run(request, "main/listar_avisos.html", {"avisos":avisos, "estado_filtro":estado_filtro, "txt_filtro":txt_filtro})
160
161 return fnc.redireccionar.login()

```

Imagen 97: Codificación del proceso principal para listar los avisos de la HUN°27

Fuente: Creación propia

ii. Resultado:

En la imagen 98 mostramos el resultado de la codificación del proceso principal para listar los avisos.

<div> <div>Listar Aviso (21)</div> <div>Todos</div> <div>Buscar...</div> <div>Buscar</div> <div></div> </div>									
Cultivo	Región	Distrito	Provincia	Sector	Evento	Creado	Resultado	Estado	Detalle
PAPA	CAJAMARCA	JAEN	CHONTALI	PACHAPIRIANA	SEQUÍA	Mar. 13, 2018, 12:48 p.m.	En Proceso	Evaluador	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	SAN PABLO	SAN LUIS	PAMPA SAN LUIS	INUNDACIÓN	Mar. 13, 2018, 12:52 p.m.	En Proceso	Responsable Región	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	JAEN	SALLIQUE	CHALANMACHE	GRANIZOS	Mar. 13, 2018, 12:52 p.m.	En Proceso	Evaluador	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	SAN MIGUEL	SAN GREGORIO	SAN GREGORIO	INUNDACIÓN	Mar. 13, 2018, 12:52 p.m.	En Proceso	Ajustador	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	SAN PABLO	SAN PABLO	JANCOS ALTO	TEMPERATURAS EXTREMAS	Mar. 13, 2018, 12:53 p.m.	En Proceso	Minagri	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	SAN PABLO	SAN PABLO	CALLANCAS	DESGLIZAMIENTO	Mar. 13, 2018, 12:53 p.m.	En Proceso	Minagri	Ver Detalle
PAPA	CAJAMARCA	CUTERVO	SAN LUIS DE LUCMA	SAIREPAMPA	HUAYCO	Mar. 13, 2018, 12:53 p.m.	En Proceso	Positiva	Ver Detalle

Imagen 98: Resultado de la Codificación de la HU N°27

Fuente: Creación propia

IV. PRUEBAS UNITARIAS DEL PROYECTO Y DE INTEGRACIÓN.

Configuración Test:

Nuestras pruebas unitarias las hemos realizado directamente de Django con la herramienta propia llamada Test User, en la imagen 99 podemos observar la configuración que hemos realizado.

```
11 class TestGeneral(unittest.TestCase):
12
13     def setUp(self):
14         self.client = Client()
15
16         if not User.objects.filter(username="testuser").exists():
17             user=User.objects.create(username='testuser', password='12345', is_staff=True, is_superuser=True)
18             user.set_password("12345")
19             user.save()
20
21         departamento=None
22         if not main.models.Departamento.objects.filter(nombre="Lima").exists():
23             departamento=main.models.Departamento.objects.create(nombre='Lima')
24         else:
25             departamento=main.models.Departamento.objects.get(nombre="Lima")
26
27         provincia=None
28         if not main.models.Provincia.objects.filter(nombre="Lima", departamento=departamento).exists():
29             provincia=main.models.Provincia.objects.create(nombre='Lima', departamento=departamento)
30         else:
31             provincia=main.models.Provincia.objects.get(nombre="Lima", departamento=departamento)
32
33         distrito=None
34         if not main.models.Distrito.objects.filter(nombre="Comas", provincia=provincia).exists():
35             distrito=main.models.Distrito.objects.create(nombre='Comas', provincia=provincia)
36         else:
37             distrito=main.models.Distrito.objects.get(nombre="Comas", provincia=provincia)
38
39         sector_estadistico=None
40         if not main.models.SectorEstadistico.objects.filter(nombre="Comas", distrito=distrito).exists():
41             sector_estadistico=main.models.SectorEstadistico.objects.create(nombre='Comas', distrito=distrito, ogr_fid=123)
42         else:
43             sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas", distrito=distrito)
44
45
```

Imagen 99: Configuración de Test - Pruebas Unitarias
Fuente: Creación propia

Test Login

En la imagen 100 hemos configurado nuestra prueba unitaria para verificar el ingreso de los usuarios al sistema web.

```
48     def test_login(self):
49         self.assertEqual(self.client.login(username="testuser",password="12345"), True)
50         response = self.client.get('/')
51         self.assertEqual("Bienvenido" in response.content, True)
52
```

Imagen 100: Prueba unitaria Login
Fuente: Creación propia

Test Registro Aviso

En la imagen 101 hemos configurado nuestra prueba unitaria para verificar el registro de avisos en el sistema web.

```

58
59 def test_registro_avisos(self):
60     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
61     departamento=main.models.Departamento.objects.get(nombre="Lima")
62     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
63     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
64     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
65
66     son_avisos=main.models.Aviso.objects.count()
67
68     response = self.client.post('/main/registrar_avisos',{
69         "departamento":departamento.id,
70         "provincia":provincia.id,
71         "distrito":distrito.id,
72         "sector_estadistico":sector_estadistico.id,
73         "cultivo":"PAPA",
74         "tipo_evento":"GRANIZOS",
75         "txt_puntos": "1****-78°25'42.3436''****-6°50'27.0957''||||2****-78°25'42.3436''****-6°53'24.5549''||||3****-78°24'42.7569''****-6°55'50.1729''||||4****-78°
76     })
77
78     self.assertEqual("Se registro con éxito" in response.content, True)
79
80
81     self.assertEqual(main.models.Aviso.objects.count() == (son_avisos + 1), True)
82
83

```

Imagen 101: Prueba Unitaria Registrar avisos

Fuente: Creación propia

Enviar Datos desde Android

En la imagen 102 hemos configurado nuestra prueba unitaria para verificar el envío de datos y sincronización desde la tablet al servidor.

```

645
646 def test_proceso_sincronizar_android_a_web(self):
647     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
648     departamento=main.models.Departamento.objects.get(nombre="Lima")
649     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
650     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
651     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
652     user=User.objects.get(username="testuser")
653
654     response = self.client.post('/main/registrar_avisos',{
655         "departamento":departamento.id,
656         "provincia":provincia.id,
657         "distrito":distrito.id,
658         "sector_estadistico":sector_estadistico.id,
659         "cultivo":"PAPA",
660         "tipo_evento":"GRANIZOS",
661         "txt_puntos": "1****-78°25'42.3436''****-6°50'27.0957''||||2****-78°25'42.3436''****-6°53'24.5549''||||3
662     })
663     aviso=main.models.Aviso.objects.filter().order_by("-id")[0]
664     aviso.f_evaluador=True
665     aviso.pago_por_factor=145
666     aviso.minimo_siembra=600
667     aviso.save()
668
669     response = self.client.post('/ajax/cargar_avisos',{
670         "i":aviso.id,
671         "d":json.dumps([
672             {"n":1, "f":1, "a":0.20, "b":0.3, "c":0.3, "d":0.4},
673             {"n":2, "f":1, "a":0.3, "b":0.3, "c":0.02, "d":0.9},
674             {"n":3, "f":1, "a":0.4, "b":0.4, "c":0.8, "d":0.87},
675             {"n":4, "f":1, "a":0.9, "b":0.9, "c":0.45, "d":0.54},
676             {"n":5, "f":1, "a":0.5, "b":0.7, "c":0.5, "d":0.36},
677             {"n":6, "f":1, "a":0.23, "b":0.5, "c":0.6, "d":0.23},
678             {"n":7, "f":1, "a":0.66, "b":0.51, "c":0.01, "d":0.1},
679             {"n":8, "f":1, "a":0.4, "b":0.31, "c":0.06, "d":0.6},
680             {"n":9, "f":1, "a":0.99, "b":0.2, "c":0.7, "d":0.5},
681             {"n":10, "f":1, "a":0.32, "b":0.11, "c":0.9, "d":0.7},
682             {"n":11, "f":1, "a":0.62, "b":0.9, "c":0.3, "d":0.4},
683         ])
684     })
685
686     aviso=main.models.Aviso.objects.filter().order_by("-id")[0]
687
688     self.assertEqual(response.status_code == 200, True)
689     self.assertEqual(json.loads(response.content)["success"] == 1, True)
690
691     self.assertEqual(aviso.total_siembra == Decimal('5.6'), True)
692     self.assertEqual(aviso.f_indemnizable , True)
693     self.assertEqual(aviso.monto_pagar == Decimal('86188') , True)
694
695

```

Imagen 102:Prueba Unitaria - Envío de datos desde Tablet

Fuente: Creación propia

Enviar Aviso de Representante a Minagri

En la imagen 103 hemos configurado nuestra prueba unitaria para verificar el envío de datos desde el usuario representante de región al MINAGRI.

```
84 def test_proceso_enviar_de_representante_a_minagri(self):
85     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
86     departamento=main.models.Departamento.objects.get(nombre="Lima")
87     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
88     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
89     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
90     user=User.objects.get(username="testuser")
91     aviso=main.models.Aviso.objects.create(
92         sector_estadistico=sector_estadistico,
93         cultivo="PAPA",
94         tipo_evento="GRANIZOS",
95         f_responsable_region=True,
96         f_minagri=False,
97         f_positiva=False,
98         f_ajustador=False,
99         f_evaluador=False,
100         f_fin=False,
101         usuario=user,
102     )
103     son_avisos_en_representante_ini=main.models.Aviso.objects.filter(
104         f_responsable_region=True,
105         f_minagri=False,
106         f_positiva=False,
107         f_ajustador=False,
108         f_evaluador=False,
109         f_fin=False,
110     ).count()
111     son_avisos_en_minagri_ini=main.models.Aviso.objects.filter(
112         f_responsable_region=False,
113         f_minagri=True,
114         f_positiva=False,
115         f_ajustador=False,
116         f_evaluador=False,
117         f_fin=False,
118     ).count()
119     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
120         "btnEnviar":"",
121         "tipo_evento":"GRANIZOS",
122         "cultivo":"PAPA",
123     })
124     self.assertEqual(response.status_code == 200, True)
125     son_avisos_en_representante_fin=main.models.Aviso.objects.filter(
126         f_responsable_region=True,
127         f_minagri=False,
128         f_positiva=False,
129         f_ajustador=False,
130         f_evaluador=False,
131         f_fin=False,
132     ).count()
133     son_avisos_en_minagri_fin=main.models.Aviso.objects.filter(
134         f_responsable_region=False,
135         f_minagri=True,
136         f_positiva=False,
137         f_ajustador=False,
138         f_evaluador=False,
139         f_fin=False,
140     ).count()
141     self.assertEqual((son_avisos_en_representante_ini - 1) == son_avisos_en_representante_fin, True)
142     self.assertEqual((son_avisos_en_minagri_ini + 1) == son_avisos_en_minagri_fin, True)
```

Imagen 103: Prueba unitaria - Envío de datos al MINAGRI
Fuente: Creación propia

Enviar Aviso de Minagri a Representante

En la imagen 104 hemos configurado nuestra prueba unitaria para verificar el retorno de datos desde el usuario MINAGRI al representante de región.

```
148 def test_proceso_enviar_de_minagri_a_representante(self):
149     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
150     departamento=main.models.Departamento.objects.get(nombre="Lima")
151     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
152     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
153     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
154     user=User.objects.get(username="testuser")
155     aviso=main.models.Aviso.objects.create(
156         sector_estadistico=sector_estadistico,
157         cultivo="PAPA",
158         tipo_evento="GRANIZOS",
159         f_responsable_region=False,
160         f_minagri=True,
161         f_positiva=False,
162         f_ajustador=False,
163         f_evaluador=False,
164         f_fin=False,
165         usuario=user,
166     )
167     son_avisos_en_representante_ini=main.models.Aviso.objects.filter(
168         f_responsable_region=True,
169         f_minagri=False,
170         f_positiva=False,
171         f_ajustador=False,
172         f_evaluador=False,
173         f_fin=False,
174     ).count()
175     son_avisos_en_minagri_ini=main.models.Aviso.objects.filter(
176         f_responsable_region=False,
177         f_minagri=True,
178         f_positiva=False,
179         f_ajustador=False,
180         f_evaluador=False,
181         f_fin=False,
182     ).count()
183
184     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
185         "btnRetornar":"" ,
186     })
187     self.assertEqual(response.status_code == 200, True)
188
189     son_avisos_en_representante_fin=main.models.Aviso.objects.filter(
190         f_responsable_region=True,
191         f_minagri=False,
192         f_positiva=False,
193         f_ajustador=False,
194         f_evaluador=False,
195         f_fin=False,
196     ).count()
197     son_avisos_en_minagri_fin=main.models.Aviso.objects.filter(
198         f_responsable_region=False,
199         f_minagri=True,
200         f_positiva=False,
201         f_ajustador=False,
202         f_evaluador=False,
203         f_fin=False,
204     ).count()
205
206     self.assertEqual((son_avisos_en_representante_fin - 1 )== son_avisos_en_representante_ini, True)
207     self.assertEqual((son_avisos_en_minagri_fin + 1 )== son_avisos_en_minagri_ini, True)
208
```

Imagen 104: Prueba unitaria- Retorno de información al Representante de Región
Fuente: Creación propia

Enviar Aviso de Minagri a Positiva

En la imagen 105 hemos configurado nuestra prueba unitaria para verificar el envío de datos desde el usuario MINAGRI a positiva.

```
210 def test_proceso_enviar_de_minagri_a_positiva(self):
211     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
212     departamento=main.models.Departamento.objects.get(nombre="Lima")
213     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
214     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
215     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
216     user=User.objects.get(username="testuser")
217     aviso=main.models.Aviso.objects.create(
218         sector_estadistico=sector_estadistico,
219         cultivo="PAPA",
220         tipo_evento="GRANIZOS",
221         f_responsable_region=False,
222         f_minagri=True,
223         f_positiva=False,
224         f_ajustador=False,
225         f_evaluador=False,
226         f_fin=False,
227         usuario=user,
228     )
229     son_avisos_en_minagri_ini=main.models.Aviso.objects.filter(
230         f_responsable_region=False,
231         f_minagri=True,
232         f_positiva=False,
233         f_ajustador=False,
234         f_evaluador=False,
235         f_fin=False,
236     ).count()
237     son_avisos_en_positiva_ini=main.models.Aviso.objects.filter(
238         f_responsable_region=False,
239         f_minagri=False,
240         f_positiva=True,
241         f_ajustador=False,
242         f_evaluador=False,
243         f_fin=False,
244     ).count()
245
246     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
247         "btnEnviar":"" ,
248     })
249     self.assertEqual(response.status_code == 200, True)
250
251     son_avisos_en_minagri_fin=main.models.Aviso.objects.filter(
252         f_responsable_region=False,
253         f_minagri=True,
254         f_positiva=False,
255         f_ajustador=False,
256         f_evaluador=False,
257         f_fin=False,
258     ).count()
259     son_avisos_en_positiva_fin=main.models.Aviso.objects.filter(
260         f_responsable_region=False,
261         f_minagri=False,
262         f_positiva=True,
263         f_ajustador=False,
264         f_evaluador=False,
265         f_fin=False,
266     ).count()
267
268     self.assertEqual((son_avisos_en_minagri_ini - 1 )== son_avisos_en_minagri_fin, True)
269     self.assertEqual((son_avisos_en_positiva_ini + 1 )== son_avisos_en_positiva_fin, True)
```

Imagen 105: Prueba unitaria - Envío de datos a Positiva
Fuente: Creación propia

Enviar Aviso de Positiva a Minagri

En la imagen 106 hemos configurado nuestra prueba unitaria para verificar el retorno de datos desde el usuario positiva al MINAGRI.

```
271
272 ▼ def test_proceso_enviar_de_positiva_a_minagri(self):
273     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
274     departamento=main.models.Departamento.objects.get(nombre="Lima")
275     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
276     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
277     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
278     user=User.objects.get(username="testuser")
279 ▼     aviso=main.models.Aviso.objects.create(
280         sector_estadistico=sector_estadistico,
281         cultivo="PAPA",
282         tipo_evento="GRANIZOS",
283         f_responsable_region=False,
284         f_minagri=False,
285         f_positiva=True,
286         f_ajustador=False,
287         f_evaluador=False,
288         f_fin=False,
289         usuario=user,
290     )
291 ▼     son_avisos_en_minagri_ini=main.models.Aviso.objects.filter(
292         f_responsable_region=False,
293         f_minagri=True,
294         f_positiva=False,
295         f_ajustador=False,
296         f_evaluador=False,
297         f_fin=False,
298     ).count()
299 ▼     son_avisos_en_positiva_ini=main.models.Aviso.objects.filter(
300         f_responsable_region=False,
301         f_minagri=False,
302         f_positiva=True,
303         f_ajustador=False,
304         f_evaluador=False,
305         f_fin=False,
306     ).count()
307
308     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
309         "btnRetornar":""})
310
311     self.assertEqual(response.status_code == 200, True)
312
313 ▼     son_avisos_en_minagri_fin=main.models.Aviso.objects.filter(
314         f_responsable_region=False,
315         f_minagri=True,
316         f_positiva=False,
317         f_ajustador=False,
318         f_evaluador=False,
319         f_fin=False,
320     ).count()
321 ▼     son_avisos_en_positiva_fin=main.models.Aviso.objects.filter(
322         f_responsable_region=False,
323         f_minagri=False,
324         f_positiva=True,
325         f_ajustador=False,
326         f_evaluador=False,
327         f_fin=False,
328     ).count()
329
330     self.assertEqual((son_avisos_en_positiva_ini - 1)== son_avisos_en_positiva_fin, True)
331     self.assertEqual((son_avisos_en_minagri_ini + 1 )== son_avisos_en_minagri_fin, True)
332
```

Imagen 106: Prueba unitaria- Retorno de información al MINAGRI

Fuente: Creación propia

Enviar Aviso de Positiva a Ajustador

En la imagen 107 hemos configurado nuestra prueba unitaria para verificar el envío de datos desde el usuario positiva al ajustador.

```
334 def test_proceso_enviar_de_positiva_a_ajustador(self):
335     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
336     departamento=main.models.Departamento.objects.get(nombre="Lima")
337     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
338     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
339     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
340     user=User.objects.get(username="testuser")
341     aviso=main.models.Aviso.objects.create(
342         sector_estadistico=sector_estadistico,
343         cultivo="PAPA",
344         tipo_evento="GRANIZOS",
345         f_responsable_region=False,
346         f_minagri=False,
347         f_positiva=True,
348         f_ajustador=False,
349         f_evaluador=False,
350         f_fin=False,
351         usuario=user,
352     )
353     son_avisos_en_ajustador_ini=main.models.Aviso.objects.filter(
354         f_responsable_region=False,
355         f_minagri=False,
356         f_positiva=False,
357         f_ajustador=True,
358         f_evaluador=False,
359         f_fin=False,
360     ).count()
361     son_avisos_en_positiva_ini=main.models.Aviso.objects.filter(
362         f_responsable_region=False,
363         f_minagri=False,
364         f_positiva=True,
365         f_ajustador=False,
366         f_evaluador=False,
367         f_fin=False,
368     ).count()
369
370     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
371         "btnEnviar":""
372     })
373     self.assertEqual(response.status_code == 200, True)
374
375     son_avisos_en_ajustador_fin=main.models.Aviso.objects.filter(
376         f_responsable_region=False,
377         f_minagri=False,
378         f_positiva=False,
379         f_ajustador=True,
380         f_evaluador=False,
381         f_fin=False,
382     ).count()
383     son_avisos_en_positiva_fin=main.models.Aviso.objects.filter(
384         f_responsable_region=False,
385         f_minagri=False,
386         f_positiva=True,
387         f_ajustador=False,
388         f_evaluador=False,
389         f_fin=False,
390     ).count()
391
392     self.assertEqual((son_avisos_en_positiva_ini - 1 )== son_avisos_en_positiva_fin, True)
393     self.assertEqual((son_avisos_en_ajustador_ini + 1 )== son_avisos_en_ajustador_fin, True)
394
```

Imagen 107: Prueba unitaria - Envío de datos al Ajustador

Fuente: Creación propia

Enviar Aviso de Ajustador a Positiva

En la imagen 108 hemos configurado nuestra prueba unitaria para verificar el retorno de datos desde el usuario ajustador a positiva.

```
395
396
397 def test_proceso_enviar_de_ajustador_a_positiva(self):
398     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
399     departamento=main.models.Departamento.objects.get(nombre="Lima")
400     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
401     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
402     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
403     user=User.objects.get(username="testuser")
404     aviso=main.models.Aviso.objects.create(
405         sector_estadistico=sector_estadistico,
406         cultivo="PAPA",
407         tipo_evento="GRANIZOS",
408         f_responsable_region=False,
409         f_minagri=False,
410         f_positiva=False,
411         f_ajustador=True,
412         f_evaluador=False,
413         f_fin=False,
414         usuario=user,
415     )
416     son_avisos_en_ajustador_ini=main.models.Aviso.objects.filter(
417         f_responsable_region=False,
418         f_minagri=False,
419         f_positiva=False,
420         f_ajustador=True,
421         f_evaluador=False,
422         f_fin=False,
423     ).count()
424     son_avisos_en_positiva_ini=main.models.Aviso.objects.filter(
425         f_responsable_region=False,
426         f_minagri=False,
427         f_positiva=True,
428         f_ajustador=False,
429         f_evaluador=False,
430         f_fin=False,
431     ).count()
432     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
433         "btnRetornar":""
434     })
435     self.assertEqual(response.status_code == 200, True)
436
437     son_avisos_en_ajustador_fin=main.models.Aviso.objects.filter(
438         f_responsable_region=False,
439         f_minagri=False,
440         f_positiva=False,
441         f_ajustador=True,
442         f_evaluador=False,
443         f_fin=False,
444     ).count()
445     son_avisos_en_positiva_fin=main.models.Aviso.objects.filter(
446         f_responsable_region=False,
447         f_minagri=False,
448         f_positiva=True,
449         f_ajustador=False,
450         f_evaluador=False,
451         f_fin=False,
452     ).count()
453
454     self.assertEqual((son_avisos_en_ajustador_ini - 1)== son_avisos_en_ajustador_fin, True)
455     self.assertEqual((son_avisos_en_positiva_ini + 1)== son_avisos_en_positiva_fin, True)
456
```

Imagen 108: Imagen 105: Prueba unitaria- Retorno de información a Positiva

Fuente: Creación propia

Enviar Aviso de Ajustador a Evaluador

En la imagen 109 hemos configurado nuestra prueba unitaria para verificar el envío de datos desde el usuario ajustador al evaluador.

```
459 def test_proceso_enviar_de_ajustador_a_evaluador(self):
460     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
461     departamento=main.models.Departamento.objects.get(nombre="Lima")
462     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
463     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
464     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
465     user=User.objects.get(username="testuser")
466     aviso=main.models.Aviso.objects.create(
467         sector_estadistico=sector_estadistico,
468         cultivo="PAPA",
469         tipo_evento="GRANIZOS",
470         f_responsable_region=False,
471         f_minagri=False,
472         f_positiva=False,
473         f_ajustador=True,
474         f_evaluador=False,
475         f_fin=False,
476         usuario=user,
477     )
478     son_avisos_en_ajustador_ini=main.models.Aviso.objects.filter(
479         f_responsable_region=False,
480         f_minagri=False,
481         f_positiva=False,
482         f_ajustador=True,
483         f_evaluador=False,
484         f_fin=False,
485     ).count()
486     son_avisos_en_evaluador_ini=main.models.Aviso.objects.filter(
487         f_responsable_region=False,
488         f_minagri=False,
489         f_positiva=False,
490         f_ajustador=False,
491         f_evaluador=True,
492         f_fin=False,
493     ).count()
494     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
495         "btnEnviar":"" ,
496     })
497     self.assertEqual(response.status_code == 200, True)
498     son_avisos_en_ajustador_fin=main.models.Aviso.objects.filter(
499         f_responsable_region=False,
500         f_minagri=False,
501         f_positiva=False,
502         f_ajustador=True,
503         f_evaluador=False,
504         f_fin=False,
505     ).count()
506     son_avisos_en_evaluador_fin=main.models.Aviso.objects.filter(
507         f_responsable_region=False,
508         f_minagri=False,
509         f_positiva=False,
510         f_ajustador=False,
511         f_evaluador=True,
512         f_fin=False,
513     ).count()
514     self.assertEqual((son_avisos_en_ajustador_ini - 1) == son_avisos_en_ajustador_fin, True)
515     self.assertEqual((son_avisos_en_evaluador_ini + 1) == son_avisos_en_evaluador_fin, True)
516
517
518
519
```

Imagen 109: Prueba unitaria - Envío de datos al Evaluador
Fuente: Creación propia

Enviar Aviso de Evaluador a Ajustador

En la imagen 110 hemos configurado nuestra prueba unitaria para verificar el retorno de datos desde el usuario evaluador al ajustador.

```
520
521 ▼ def test_proceso_enviar_de_evaluador_a_ajustador(self):
522     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
523     departamento=main.models.Departamento.objects.get(nombre="Lima")
524     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
525     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
526     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
527     user=User.objects.get(username="testuser")
528 ▼     aviso=main.models.Aviso.objects.create(
529         sector_estadistico=sector_estadistico,
530         cultivo="PAPA",
531         tipo_evento="GRANIZOS",
532         f_responsable_region=False,
533         f_minagri=False,
534         f_positiva=False,
535         f_ajustador=False,
536         f_evaluador=True,
537         f_fin=False,
538         usuario=user,
539     )
540 ▼     son_avisos_en_ajustador_ini=main.models.Aviso.objects.filter(
541         f_responsable_region=False,
542         f_minagri=False,
543         f_positiva=False,
544         f_ajustador=True,
545         f_evaluador=False,
546         f_fin=False,
547     ).count()
548 ▼     son_avisos_en_evaluador_ini=main.models.Aviso.objects.filter(
549         f_responsable_region=False,
550         f_minagri=False,
551         f_positiva=False,
552         f_ajustador=False,
553         f_evaluador=True,
554         f_fin=False,
555     ).count()
556
557     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
558         "btnRetornar": "",
559     })
560     self.assertEqual(response.status_code == 200, True)
561
562 ▼     son_avisos_en_ajustador_fin=main.models.Aviso.objects.filter(
563         f_responsable_region=False,
564         f_minagri=False,
565         f_positiva=False,
566         f_ajustador=True,
567         f_evaluador=False,
568         f_fin=False,
569     ).count()
570 ▼     son_avisos_en_evaluador_fin=main.models.Aviso.objects.filter(
571         f_responsable_region=False,
572         f_minagri=False,
573         f_positiva=False,
574         f_ajustador=True,
575         f_evaluador=True,
576         f_fin=False,
577     ).count()
578
579     self.assertEqual((son_avisos_en_evaluador_ini - 1 )== son_avisos_en_evaluador_fin, True)
580     self.assertEqual((son_avisos_en_ajustador_ini + 1 )== son_avisos_en_ajustador_fin, True)
581
```

Imagen 110:Prueba unitaria- Retorno de información al Ajustador
Fuente: Creación propia

Enviar Aviso de Evaluador a Terminar

En la imagen 111 mostramos la configuración que se ha realizado de nuestra prueba unitaria para verificar el fin de una evaluación desde la web.

```
583 def test_proceso_enviar_de_evaluador_a_fin(self):
584     self.assertEqual(self.client.login(username="testuser",password="12345"), True)
585     departamento=main.models.Departamento.objects.get(nombre="Lima")
586     provincia=main.models.Provincia.objects.get(nombre="Lima",departamento=departamento)
587     distrito=main.models.Distrito.objects.get(nombre="Comas",provincia=provincia)
588     sector_estadistico=main.models.SectorEstadistico.objects.get(nombre="Comas",distrito=distrito)
589     user=User.objects.get(username="testuser")
590     aviso=main.models.Aviso.objects.create(
591         sector_estadistico=sector_estadistico,
592         cultivo="PAPA",
593         tipo_evento="GRANIZOS",
594         f_responsable_region=False,
595         f_minagri=False,
596         f_positiva=False,
597         f_ajustador=False,
598         f_evaluador=True,
599         f_fin=False,
600         usuario=user,
601     )
602     son_avisos_en_evaluador_ini=main.models.Aviso.objects.filter(
603         f_responsable_region=False,
604         f_minagri=False,
605         f_positiva=False,
606         f_ajustador=False,
607         f_evaluador=True,
608         f_fin=False,
609     ).count()
610     son_avisos_en_fin_ini=main.models.Aviso.objects.filter(
611         f_responsable_region=False,
612         f_minagri=False,
613         f_positiva=False,
614         f_ajustador=False,
615         f_evaluador=False,
616         f_fin=True,
617     ).count()
618
619     response = self.client.post('/main/detalle_aviso/%s/' % aviso.id,{
620         "btnCerrarEvaluacion":""})
621
622     self.assertEqual(response.status_code == 200, True)
623
624     son_avisos_en_evaluador_fin=main.models.Aviso.objects.filter(
625         f_responsable_region=False,
626         f_minagri=False,
627         f_positiva=False,
628         f_ajustador=False,
629         f_evaluador=True,
630         f_fin=False,
631     ).count()
632     son_avisos_en_fin_fin=main.models.Aviso.objects.filter(
633         f_responsable_region=False,
634         f_minagri=False,
635         f_positiva=False,
636         f_ajustador=False,
637         f_evaluador=False,
638         f_fin=True,
639     ).count()
640     self.assertEqual((son_avisos_en_evaluador_ini - 1 )== son_avisos_en_evaluador_fin, True)
641     self.assertEqual((son_avisos_en_fin_ini + 1 )== son_avisos_en_fin_fin, True)
642
```

Imagen 111: Prueba Unitaria - Terminar aviso

Fuente: Creación propia

Resultados TestUnit y las pruebas de integración

Cuando se programa usando Python y Django las pruebas unitarias se verifican emitiendo un solo resultado, en la imagen 112 nos muestra el resultado del test de pruebas automatizadas que indica que hay 0 errores, es decir, todos los procesos principales funcionan correctamente.

Podemos afirmar que nuestras pruebas de integración teniendo como objetivo verificar el correcto funcionamiento entre los distintos componentes una vez que han sido probados unitariamente, interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales, como lo vemos en la imagen 111, la cual es el resultado de las pruebas unitarias funcionando correctamente con los demás componentes.

```
(venv)root@myserverone:/home/tesis/code/tesis/catastrofico# python manage.py test
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 12 tests in 4.025s

OK
Destroying test database for alias 'default'...
```

Imagen 112: Resultado de las pruebas unitarias

Fuente: Creación Django

V. RESULTADOS:

Nuestros resultados lo hemos expresado en la tabla N° 60, en la cual mostramos nuestro comparativo respecto a cómo funcionaban los procesos antes de usar el sistema y cómo funcionan ahora:

EVENTO	ANTES	AHORA
Dar a conocer a La Positiva los incidentes que serán atendidos por el seguro catastrófico.	Los representantes de región enviaban a positiva los incidentes con un desfase de 15 días	Los representantes de región registran los incidentes en el sistema y ya tienen conocimiento en forma automática e inmediata positiva. Actualmente ingresan los incidentes a más tardar en 2 días .
La Positiva debe avisar de los incidentes a la empresa de servicios de Peritaje Agrario (Agroevaluaciones) para su atención.	Positiva le hacía llegar la información mediante correo, lo cual le demoraba 10 días terminar de enviar todos los incidentes que se deben evaluar.	Cuando los representantes de región a medida que van registrando los incidentes, ya están informados positiva, MINAGRI y agroevaluaciones. Lo cual les demora 2 días para terminar de hacer llegar la información.
Agroevaluaciones tiene que atender en campo los avisos de siniestros registrados.	La programación para inspección en campo se daba después de que La positiva terminaba de enviar los incidentes, recién se podía realizar las programaciones, es decir demoraban en total 10 días en total. Ya que tenían que informar a sus respectivos ajustadores de cada región cuales son los avisos que deben atender.	Al tener todos los avisos registrados en el sistema, este los divide por regiones y automáticamente ya el ajustador de cada región asegurada ya sabe que avisos debe atender y proceder con su programación. Lo cual lo hacen en un tiempo de 3 días (dos días para esperar que todos los avisos sean ingresados por parte de los representantes de región y un día para programar todas sus inspecciones).
Los evaluadores tienen que emitir sus actas a La Positiva con el dictamen de los avisos (desestimado, indemnizable y no indemnizable)	Se demoraban 15 días a más en hacer llegar las actas con la firma del representante de región, verificando que están conforme con la inspección a positiva.	Las actas son emitidas el mismo día de la inspección, a más tardar demora 1 día para que las actas ya estén reflejadas en el sistema y pueda visualizar positiva, Minagri y los representantes de región.
TOTAL, DE DÍAS:	50 días	8 días

Tabla 60: Análisis de productividad del sistema
Fuente: creación Propia a través de la información brindada por el cliente

CAPÍTULO IV: CONCLUSIONES

Por lo cual nuestras conclusiones respecto al comparativo de la tabla anterior, podemos indicar lo siguiente:

- El desarrollo del sistema de Gestión, Análisis y Cartografía para el seguro agrícola catastrófico logró minimizar en un 84% los tiempos de respuesta de toda la información necesaria para la toma de decisiones de todos los usuarios involucrados de una manera oportuna, relevante y confiable.
- Se planificó el desarrollo del sistema estableciendo 8 iteraciones y 27 historias de usuario, tomando en cuenta las necesidades de las empresas involucrados en el proceso, analizando todos los escenarios posibles en los que se puedan encontrar cada usuario, logrando así optimizar recursos y tener en claro cuál es el proceso que debe seguir el sistema.
- Se diseñaron e implementaron una interfaz intuitiva de fácil predicción la cual fue aceptada por todos los usuarios involucrados.
- Mediante la codificación de nuestro sistema hemos logrado optimizar recursos como es el tiempo de atención y soluciones inmediatas, así mismo los recursos humanos.
- Se realizaron las pruebas unitarias y de integración garantizando así la confiabilidad del sistema, se validaron los procesos principales lo cual nos permitirán controlar de manera oportuna algún error que pueda más existir con un desarrollo posterior. Logrando así que las entidades pertinentes, estén seguros de que la información está protegida, es confidencial y funciona correctamente.

CAPÍTULO V: RECOMENDACIONES

Las recomendaciones para que el sistema sea aún más efectivo es que:

- Integrar firmas digitales para que las actas de conformidad ya no sean impresas y sea enviado de una manera más oportuna y segura.
- Complementar las inspecciones en campo con las firmas espectrales a través del satélite, el cual brindará un indicativo que ayudará a determinar el estado del cultivo de manera inmediata.
- Incluir el uso drones para inspección en campo, buscando optimizar el tiempo invertido en cada inspección.
- Desarrollar un modelo de predicción que permita identificar cuáles son las zonas de mayor riesgo de fenómenos naturales y provocados por el hombre.

BIBLIOGRAFÍA REFERENCIADA

REFERENCIAS

Agroevaluaciones, P. (2019). Agroevaluaciones Peru. Obtenido de Información General:
<http://agroevaluacionesperu.com/>

Benbourahla, N. (2015). Android 5: Principios del desarrollo de aplicaciones Java.
Barcelona: Ediciones ENI.

Castillo , O., Figueroa , D., & Sevilla , H. (2019). Programación Extrema. Obtenido de Fases
XP: <http://programacionextrema.tripod.com/fases.htm>

Cunningham, W. (2001). Ágil Manifiesto. Obtenido de Manifiesto para el desarrollo ágil de
software: <http://agilemanifesto.org/>

Gutierrez, E. (2009). • JavaScript: Conceptos básicos y avanzados (bibliotecas Prototype y
Script.aculo.us). EDICIONES ENI.

Holovaty, A., & Kaplan , J. (2009). La guia definitiva de Django. Anaya Multimedia.

MINAGRI. (2019). MINAGRI. Obtenido de ¿Qué es el seguro agrícola catastrófico?:
http://minagri.gob.pe/portal/download/pdf/programasespeciales/seguro_agropecuario/seguro_agropecuario.pdf

MINAGRI. (2019). Seguro Agrícola Catastrófico. Obtenido de Triptico:
http://minagri.gob.pe/portal/download/pdf/programasespeciales/seguro_agropecuario/triptico.sac.pdf

ORACLE. (2019). MySQL. Obtenido de Manual de referencia de MySQL 8.0:
<https://dev.mysql.com/doc/refman/8.0/en/>

- Peters, T. (04 de junio de 1999). Recursos Python. Obtenido de El Zen de Python:
<https://recursospython.com/guias-y-manuales/zen-de-python-tim-peters/>
- Positiva, S. (2019). La Positiva Seguros. Obtenido de Información General:
<https://www.lapositiva.com.pe/wps/portal/corporativo/home/nosotros#conocenos>
- Tomlin, C. D. (1990). Sistemas de información geográfica y modelización cartográfica. New Jersey: Prentice-Hall Inc.

TRABAJOS CITADOS

- Juan, C. (2009). Repositorio de la UNIA. Obtenido de Elaboración de cartografía de riesgo de inundaciones y propuesta de mejora de conservación de la cuenca y reserva MAB de Mar Chiquita, provincia de Buenos Aires, Argentina:
<http://hdl.handle.net/10334/84>
- Ligia, P. (Agosto de 2016). Repositorio Digital USFQ. Obtenido de Rol de los Sistemas de Información Geográfica (SIG) dentro de un plan de ordenamiento territorial y desarrollo a nivel cantonal con un enfoque geo-administrativo:
<http://repositorio.usfq.edu.ec/handle/23000/6193>
- Mario, B. (Agosto de 2011). Repositorio digital de tesis PUCP. Obtenido de Desarrollo de un módulo SIG para el manejo de Imágenes multiespectrales orientado a la agricultura de precisión:
<http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/1132>

ANEXOS

I. Anexo 1: Entendimiento del Negocio:

Nuestro historial de planificación y desarrollo del sistema lo podemos observar en la tabla n°61, que mostramos a continuación:

Versión	Autor	Resumen de los Cambios	Fecha
0.0.1	Christian Rodas Cinthya Amaya	Realización de Flujo de trabajo	12/04/17
1.0	POSITIVA Y AGROEVALUACIONES	Revisión de Flujo de trabajo	15/01/2018
2.0	POSITIVA Y AGROEVALUACIONES	Aprobación de Flujo de trabajo	25/03/2018
1.0	POSITIVA Y AGROEVALUACIONES y MINAGRI	Revisión del funcionamiento del sistema	03/07/2018
2.0	POSITIVA Y AGROEVALUACIONES y MINAGRI	Aprobación del funcionamiento del sistema	22/08/2018

**Tabla 61: Historial de planificación y desarrollo del sistema.
Fuente: Creación propia**

a. JUSTIFICACIÓN DEL MACROPROCESO BAJO ESTUDIO

➤ Definamos la interfaz de nuestro sistema:

Como podemos observar en la imagen 113, detallamos la interfaz de nuestro sistema web.

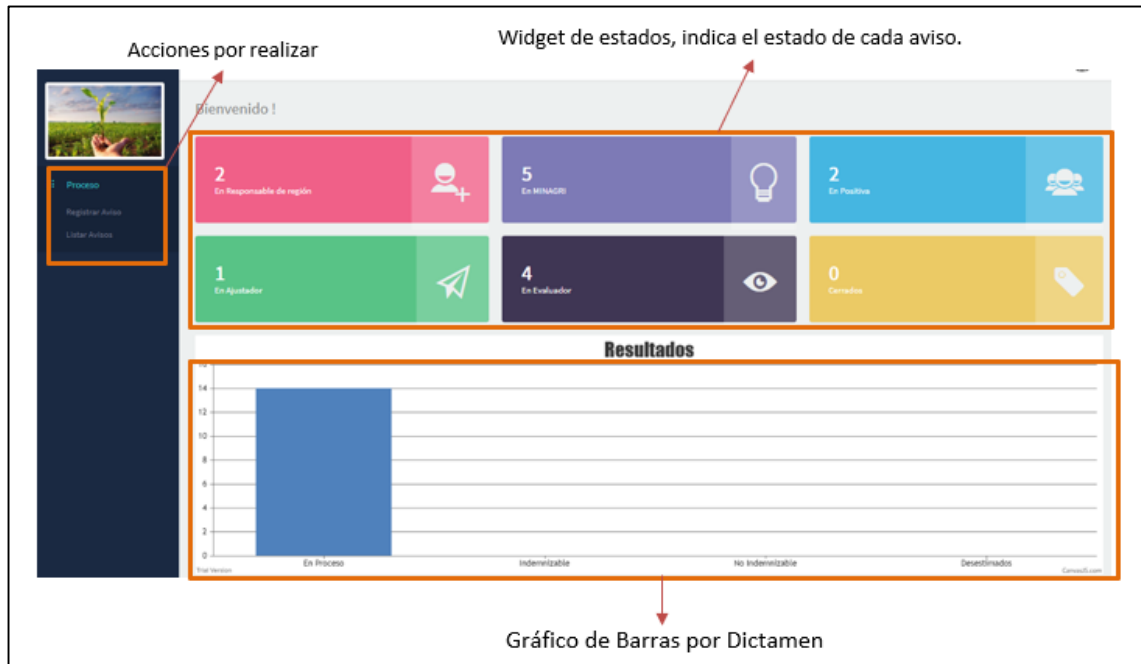


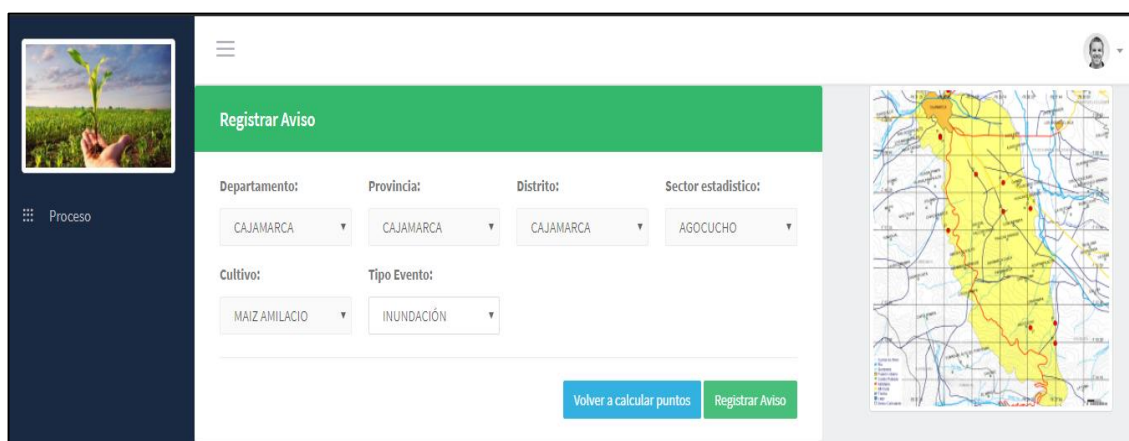
Imagen 113: Interfaz del Sistema

Fuente: Creación propia

➤ Definamos los estados

✓ Estado Ingresado:

Este usuario (DRA) se encargará de ingresar los avisos de siniestros notificados por el agente agrario de cada región. Lo campos que ingresará como lo muestra la imagen N° 114, son los siguientes:



Registrar Aviso

Departamento: CAJAMARCA Provincia: CAJAMARCA Distrito: CAJAMARCA Sector estadístico: AGOCUCHO

Cultivo: MAIZ AMILACIO Tipo Evento: INUNDACIÓN

Volver a calcular puntos Registrar Aviso

Imagen 114: Ingreso de datos al sistema – RR
Fuente: Creación propia

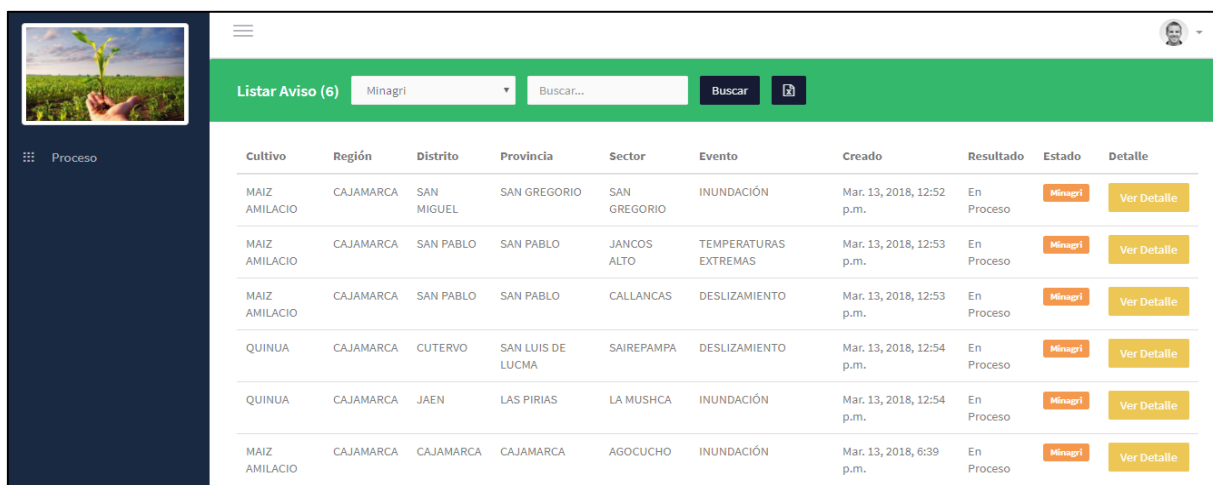
El Responsable de Región volverá a calcular puntos para que se realice la inspección, también se le mostrará el mapa de la región en la que está registrando el siniestro.

Al registrar el aviso, la información será enviada a MINAGRI. Estos avisos pueden ser visualizados por las empresas Agroevaluaciones, La Positiva y MINAGRI (los cuales solo podrán ver más no editar ningún campo).

✓ **Estado Revisado:**

Acá el aviso ya se encuentra en conocimiento del MINAGRI, quien revisará si la información ha sido ingresada correctamente, de lo contrario podrá retornarlo al responsable

Región. Los avisos se le mostrarán en una lista como se muestra en la imagen n° 115 a continuación:

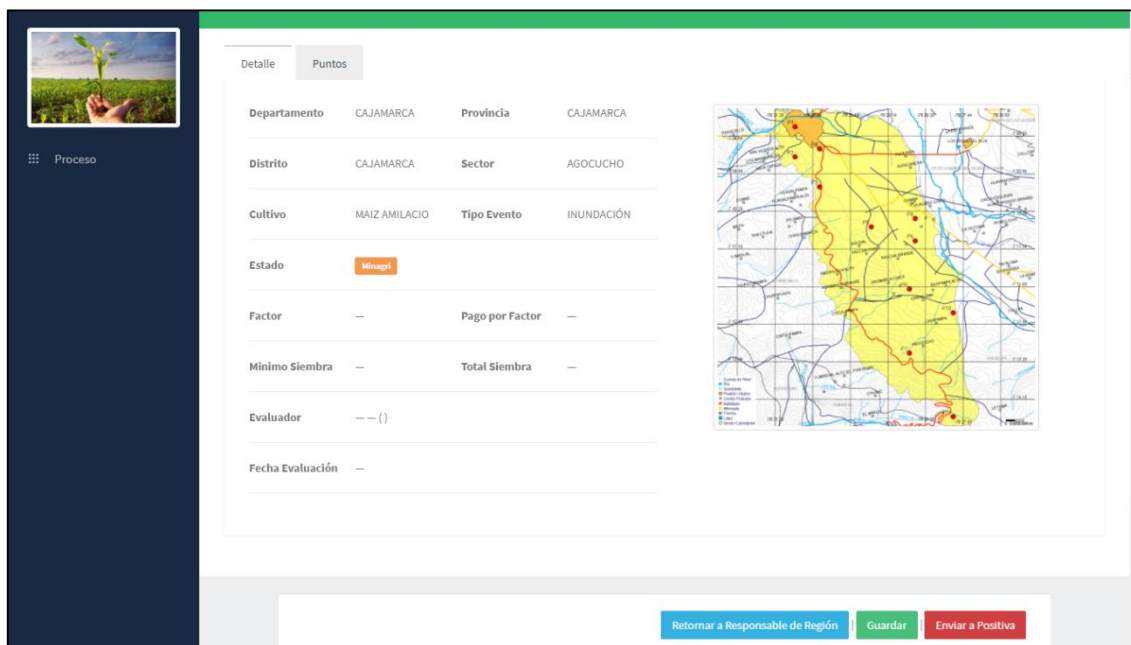


Cultivo	Región	Distrito	Provincia	Sector	Evento	Creado	Resultado	Estado	Detalle
MAIZ AMILACIO	CAJAMARCA	SAN MIGUEL	SAN GREGORIO	SAN GREGORIO	INUNDACIÓN	Mar. 13, 2018, 12:52 p.m.	En Proceso	Minagri	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	SAN PABLO	SAN PABLO	JANCOS ALTO	TEMPERATURAS EXTREMAS	Mar. 13, 2018, 12:53 p.m.	En Proceso	Minagri	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	SAN PABLO	SAN PABLO	CALLANCAS	DESGLIZAMIENTO	Mar. 13, 2018, 12:53 p.m.	En Proceso	Minagri	Ver Detalle
QUINUA	CAJAMARCA	CUTERVO	SAN LUIS DE LUCMA	SAIREPAMPA	DESGLIZAMIENTO	Mar. 13, 2018, 12:54 p.m.	En Proceso	Minagri	Ver Detalle
QUINUA	CAJAMARCA	JAEN	LAS PIRIAS	LA MUSHCA	INUNDACIÓN	Mar. 13, 2018, 12:54 p.m.	En Proceso	Minagri	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	CAJAMARCA	CAJAMARCA	AGOCUCHO	INUNDACIÓN	Mar. 13, 2018, 6:39 p.m.	En Proceso	Minagri	Ver Detalle

Imagen 115: Bandeja de avisos ingresados del MINAGRI

Fuente: creación propia

Ingresa al aviso a través del botón VER DETALLE y se mostrará los atributos de los avisos, como lo muestra la imagen 116.:



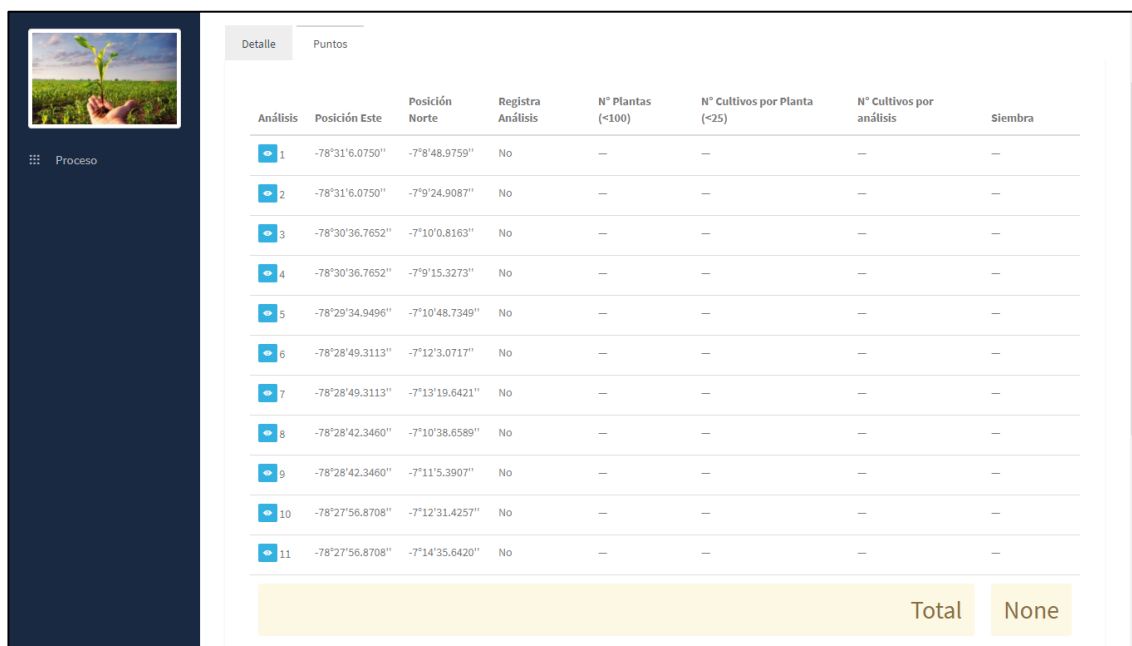
Detalle		Puntos	
Departamento	CAJAMARCA	Provincia	CAJAMARCA
Distrito	CAJAMARCA	Sector	AGOCUCHO
Cultivo	MAIZ AMILACIO	Tipo Evento	INUNDACIÓN
Estado	Minagri		
Factor	—	Pago por Factor	—
Mínimo Siembra	—	Total Siembra	—
Evaluador	— ()		
Fecha Evaluación	—		

Retornar a Responsable de Región Guardar Enviar a Positiva

Imagen 116: Detalle de Aviso – MINAGRI

Fuente: Creación propia

También puede ver los puntos de evaluación, como se muestra en la imagen n°117:



Análisis	Posición Este	Posición Norte	Registra Análisis	N° Plantas (<100)	N° Cultivos por Planta (<25)	N° Cultivos por análisis	Siembra
1	-78°31'6.0750"	-7°8'48.9759"	No	—	—	—	—
2	-78°31'6.0750"	-7°9'24.9087"	No	—	—	—	—
3	-78°30'36.7652"	-7°10'0.8163"	No	—	—	—	—
4	-78°30'36.7652"	-7°9'15.3273"	No	—	—	—	—
5	-78°29'34.9496"	-7°10'48.7349"	No	—	—	—	—
6	-78°28'49.3113"	-7°12'3.0717"	No	—	—	—	—
7	-78°28'49.3113"	-7°13'19.6421"	No	—	—	—	—
8	-78°28'42.3460"	-7°10'38.6589"	No	—	—	—	—
9	-78°28'42.3460"	-7°11'5.3907"	No	—	—	—	—
10	-78°27'56.8708"	-7°12'31.4257"	No	—	—	—	—
11	-78°27'56.8708"	-7°14'35.6420"	No	—	—	—	—
						Total	None

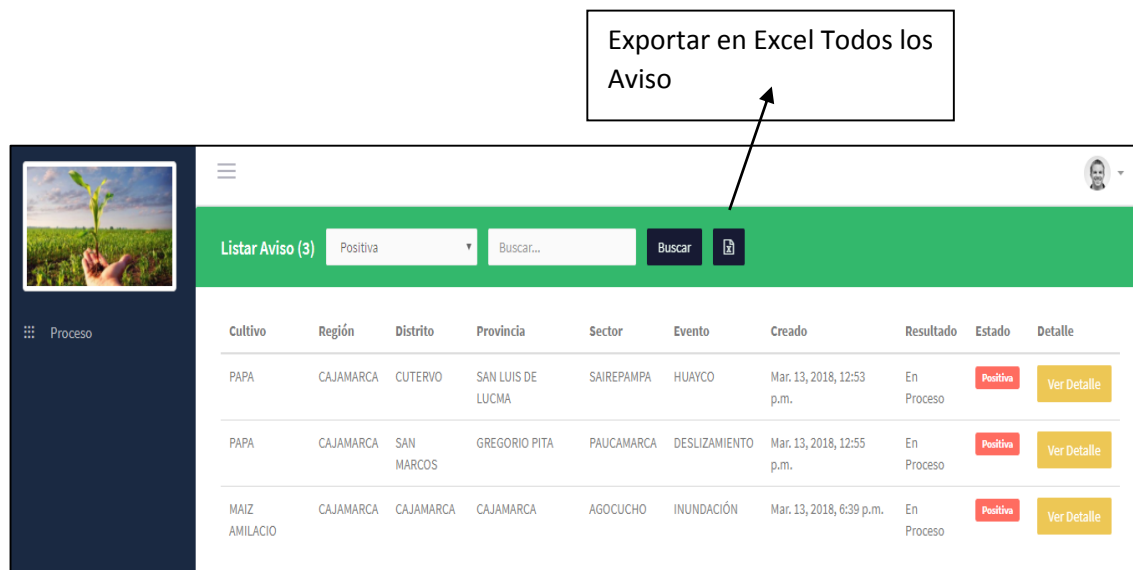
Imagen 117: Puntos a evaluar – MINAGRI
Fuente: Creación propia

Si todo está correcto, envía la información a POSITIVA.

✓ Estado Agregado:

Positiva recibe los avisos como se muestra en la imagen n° 118:

Exportar en Excel Todos los Aviso



Cultivo	Región	Distrito	Provincia	Sector	Evento	Creado	Resultado	Estado	Detalle
PAPA	CAJAMARCA	CUTERVO	SAN LUIS DE LUCMA	SAIREPAMPA	HUAYCO	Mar. 13, 2018, 12:53 p.m.	En Proceso	Positiva	Ver Detalle
PAPA	CAJAMARCA	SAN MARCOS	GREGORIO PITA	PAUCAMARCA	DESLIZAMIENTO	Mar. 13, 2018, 12:55 p.m.	En Proceso	Positiva	Ver Detalle
MAIZ AMILACIO	CAJAMARCA	CAJAMARCA	CAJAMARCA	AGOCUCHO	INUNDACIÓN	Mar. 13, 2018, 6:39 p.m.	En Proceso	Positiva	Ver Detalle

Imagen 118: Bandeja de avisos - La Positiva
Fuente: Creación propia

Agrega los siguientes datos de forma manual como se muestra en la imagen n°119, los cuales solo ellos tienen conocimiento:

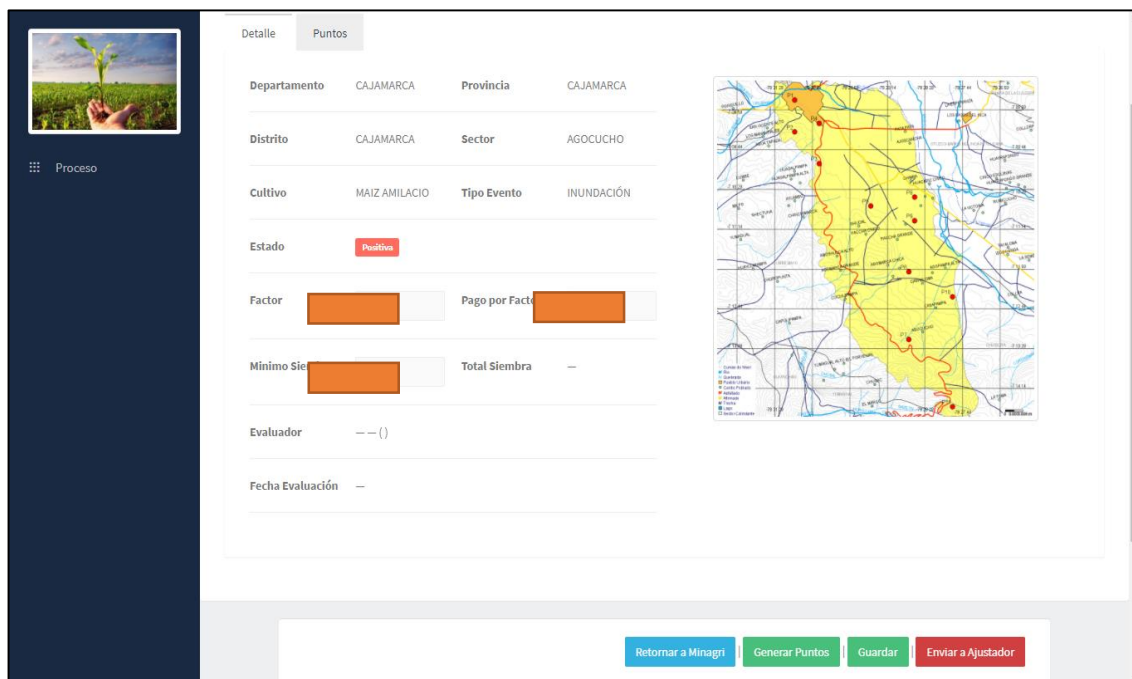


Imagen 119: Agregar datos - La Positiva
Fuente: Creación propia

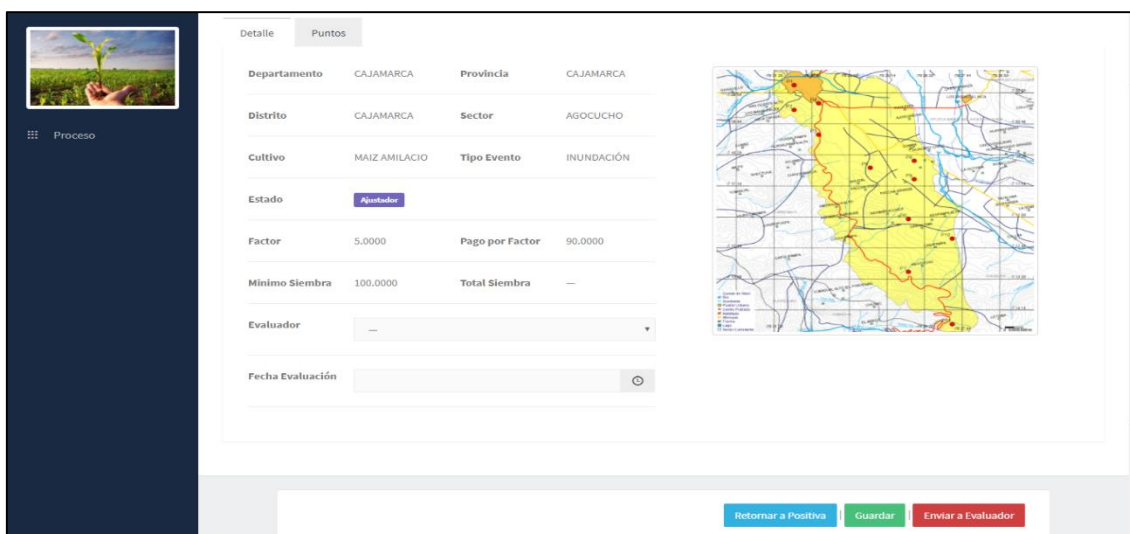
Hay que considerar que ellos como La Positiva también pueden visualizar que puntos van a ser evaluados, como se mostró en la imagen n° 117.


Una vez ingresados los datos, procede a enviar al AJUSTADOR:

✓ **Estado Asignado:**

El Ajustador recibe la información de Positiva, revisa y si todos los campos enviados están correctos entonces, procede asignar una fecha de evaluación y un evaluador al aviso, tal cual lo indica la imagen n° 120.

Una vez asignado la fecha y el evaluador, este aviso será enviado al evaluador asignado, quien podrá acceder desde la tablet.



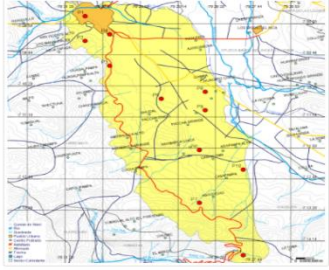


Proceso

Detalle

Puntos

Departamento	CAJAMARCA	Provincia	CAJAMARCA
Distrito	CAJAMARCA	Sector	AGOCUCHO
Cultivo	MAIZ AMILACIO	Tipo Evento	INUNDACIÓN
Estado	Ajustador		
Factor	5.0000	Pago por Factor	90.0000
Mínimo Siembra	100.0000	Total Siembra	—
Evaluador	—		
Fecha Evaluación	—		



Retornar a Positiva
Guardar
Enviar a Evaluador

Imagen 120: Asignar datos de aviso – Ajustador
Fuente: Creación propia

✓ Estado Evaluado:

Los evaluadores ya con sus avisos programados proceden a realizar la inspección de evaluación necesaria, lo cual se apoyarán de una tablet, en la que registrarán los datos solicitados según el acta que se esté trabajando. Al momento de hacer la inspección en el sector estadístico los evaluadores por cada punto de evaluación tendrán que tomar una foto a la parcela.

Al momento de ingresar al sistema de la tablet, como muestra nuestra imagen n° 121 la bandeja de avisos:

N°	Codigo	Cultivo	Departamento	Provincia	Distrito	Sector	Evento	Estado	Factor Planta	D
1	1	PAPA	CAJAMARCA	JAEN	CHONTALI	PACHAPIRIANA	SEQUIA	pendiente	0.123	No
2	2	ARVERJA	CAJAMARCA	SAN MIGUEL	EL PRADO	EL PRADO	DESIZAMIENTO	pendiente	0.05	No
3	3	QUINUA	CAJAMARCA	SAN MARCOS	EDUARDO VILLANUEVA	HUACACORRAL	TEMPERATURAS EXTREMAS	pendiente	0.023	No
4	4	MAIZ AMILACIO	CAJAMARCA	SAN PABLO	TUMBADEN	INGATAMBO	HUAYCO	pendiente	0.526	No
5	5	MAIZ AMILACIO	CAJAMARCA	CAJAMARCA	CAJAMARCA	AGOCUCHO	INUNDACION	pendiente	5	No

Imagen 121: Bandeja de avisos recibidos en tablet – Evaluador
Fuente: Creación propia

Cuando ya tengan todos sus avisos, solo tocan el aviso que desean evaluar, ahí deciden si van a evaluar el aviso o DESESTIMAR, como se muestra en la imagen n° 122, Si este último fuese el caso el proceso ha terminado y se genera el acta.

Caso contrario deben evaluar y de acuerdo esos resultados el sistema determinará si es INDEMNIZABLE o NO INDEMNIZABLE, dando por concluido el proceso.

TesisApp

ATRÁS
GUARDAR
DESESTIMAR
TERMINAR

Departamento CAJAMARCA Provincia CAJAMARCA Distrito CAJAMARCA Sector AGOCUCHO Cultivo MAIZ AMILICIO
 Evento INUNDACIÓN Factor 5

N°	Este	Norte	RqA	N° Plantas	Cultivos por Planta	Cultivos por Analisis	Siembra	Foto
1	-78°31'6.0750"	-7°8'48.9759"	<input type="checkbox"/>					FOTO 1
2	-78°31'6.0750"	-7°9'24.9087"	<input type="checkbox"/>					FOTO 2
3	-78°30'36.7652"	-7°10'0.8163"	<input type="checkbox"/>					FOTO 3
4	-78°30'36.7652"	-7°9'15.3273"	<input type="checkbox"/>					FOTO 4
5	-78°29'34.9496"	-7°10'48.7349"	<input type="checkbox"/>					FOTO 5
6	-78°28'49.3113"	-7°12'3.0717"	<input type="checkbox"/>					FOTO 6
7	-78°28'49.3113"	-7°13'19.6421"	<input type="checkbox"/>					FOTO 7

Imagen 122: Evaluar o desestimar aviso en campo

Fuente: Creación propia

Como vemos en la imagen n° 123, para una mejor ubicación del cultivo, el mapa se mostrará debajo de toda la información que se indica en la imagen anterior:

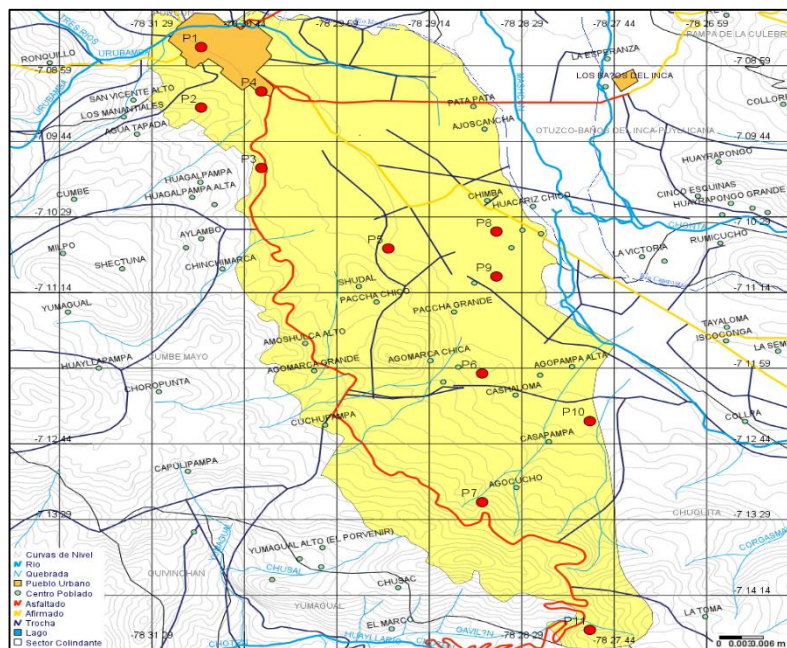
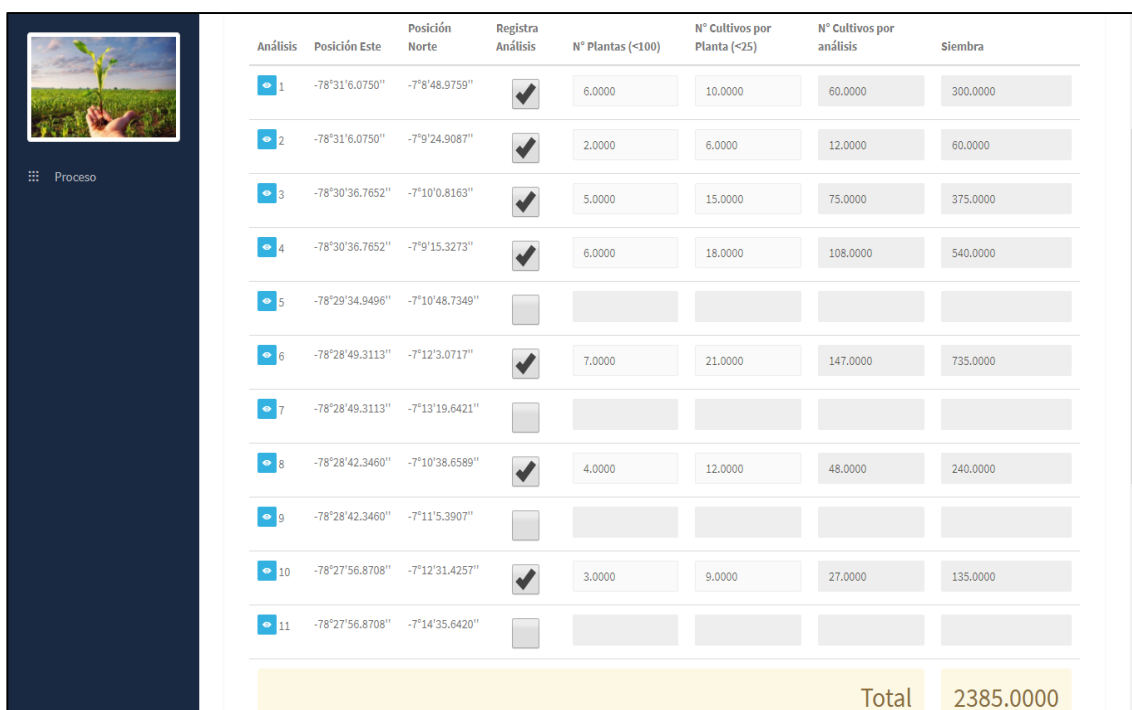


Imagen 123: Mapa que se muestra en la tablet

Fuente: Creación propia

Al terminar de evaluar los avisos en la tablet tienen que enviarlos al servidor usando un punto de internet.

Si necesitan hacer alguna observación o modificación, ya puede hacer desde la web, como muestra la imagen n° 124:

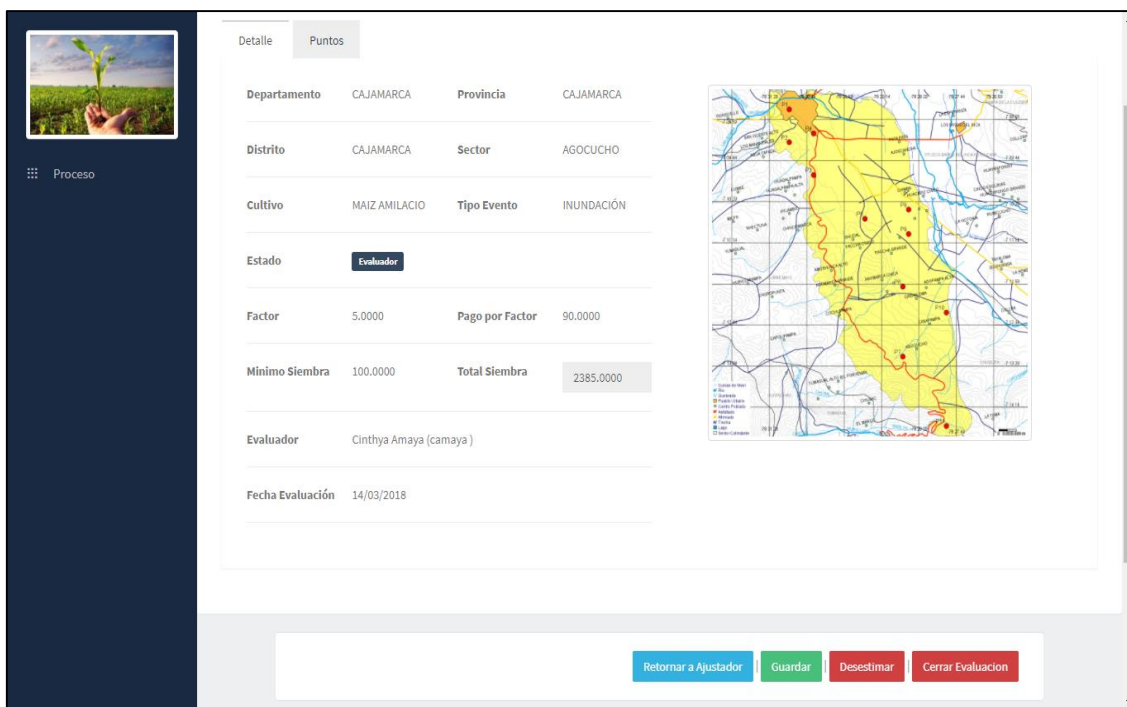


Análisis	Posición Este	Posición Norte	Registra Análisis	N° Plantas (<100)	N° Cultivos por Planta (<25)	N° Cultivos por análisis	Siembra
1	-78°31'6.0750"	-7°8'48.9759"	<input checked="" type="checkbox"/>	6.0000	10.0000	60.0000	300.0000
2	-78°31'6.0750"	-7°9'24.9087"	<input checked="" type="checkbox"/>	2.0000	6.0000	12.0000	60.0000
3	-78°30'36.7652"	-7°10'0.8163"	<input checked="" type="checkbox"/>	5.0000	15.0000	75.0000	375.0000
4	-78°30'36.7652"	-7°9'15.3273"	<input checked="" type="checkbox"/>	6.0000	18.0000	108.0000	540.0000
5	-78°29'34.9496"	-7°10'48.7349"	<input type="checkbox"/>				
6	-78°28'49.3113"	-7°12'3.0717"	<input checked="" type="checkbox"/>	7.0000	21.0000	147.0000	735.0000
7	-78°28'49.3113"	-7°13'19.6421"	<input type="checkbox"/>				
8	-78°28'42.3460"	-7°10'38.6589"	<input checked="" type="checkbox"/>	4.0000	12.0000	48.0000	240.0000
9	-78°28'42.3460"	-7°11'5.3907"	<input type="checkbox"/>				
10	-78°27'56.8708"	-7°12'31.4257"	<input checked="" type="checkbox"/>	3.0000	9.0000	27.0000	135.0000
11	-78°27'56.8708"	-7°14'35.6420"	<input type="checkbox"/>				
Total							2385.0000

Imagen 124: Revisión de avisos evaluados enviados al servidor – Evaluadores
Fuente: Creación propia

✓ **Estado terminado:**

Una vez ya corregido y revisado, los evaluadores proceden a cerrar la evaluación, como se muestra en la imagen n° 125 hay un botón rojo que indica esa acción, ya todos los usuarios pueden descargar el PDF del acta de aviso.



The screenshot displays a web application interface for evaluation management. On the left is a dark sidebar with a 'Proceso' menu. The main content area features a 'Detalle' tab and a form with the following fields:

Detalle	
Departamento	CAJAMARCA
Provincia	CAJAMARCA
Distrito	CAJAMARCA
Sector	AGOCUCHO
Cultivo	MAIZ AMILACIO
Tipo Evento	INUNDACIÓN
Estado	Evaluador
Factor	5.0000
Pago por Factor	90.0000
Mínimo Siembra	100.0000
Total Siembra	2385.0000
Evaluador	Cinthya Amaya (camaya)
Fecha Evaluación	14/03/2018

At the bottom right, there are four buttons: 'Reformar a Ajustador' (blue), 'Guardar' (green), 'Desestimar' (red), and 'Cerrar Evaluacion' (red). A map of the region is visible on the right side of the form.

Imagen 125: Cerrar evaluación – Evaluadores
Fuente: Creación propia

II. Anexo 2: Factibilidad del Proyecto:

Mediante el análisis que vamos a realizar; podremos determinar la factibilidad de nuestro sistema, considerando que el sistema es hecho para el cliente Agroevaluaciones, en la tabla 62 mostraremos el flujo de caja actual de la empresa y en la tabla 63 mostraremos el flujo de caja cuando ya cuenten con el sistema agrícola catastrófico.

Conozcamos qué significado tiene las variables que se utilizan en la factibilidad económica de un proyecto:

- **Valor Actual Neto (VAN):** El VAN de una inversión, se deduce que es la suma de los montos actualizados de todos los flujos netos de caja esperados del proyecto, deducido el valor de la inversión inicial. Si el proyecto de inversión tiene un VAN positivo, el proyecto es eficaz. Entre dos o más proyectos, el más beneficioso es el que tenga un VAN más alto.
- **Tasa Interna de Rentabilidad (TIR):** El TIR es la tasa de descuento que hace que el Valor Actual Neto (VAN.) de una inversión sea igual a cero. Este procedimiento se piensa que una inversión es aconsejable si la TIR. resultante es igual o superior a la tasa exigida por el inversor, y entre varias opciones, la más provechosa será aquella que ofrezca una TIR. mayor.
- **Costo – Beneficio:** Es el cociente que resulta de dividir el valor actual de los beneficios y el valor actual de los costos del proyecto, el cual si es mayor que 1 (uno) entonces es aceptable, si es igual que 1(unos) es indiferente y si es menor que 1(unos) se rechaza el proyecto.

1. **FLUJO DE CAJA ECONÓMICO (SIN IMPLEMENTACIÓN DEL SISTEMA AGRÍCOLA CATASTRÓFICO):**

Como se explica en la tabla 62, ese es el flujo de caja económico de Agroevaluaciones Perú, en la realidad actual, es decir sin un sistema de gestión, análisis y cartografía para el seguro agrícola catastrófico.

FLUJO DE CAJA ECONÓMICO (SIN IMPLEMENTACIÓN DEL SISTEMA AGRÍCOLA CATASTRÓFICO)						
CONCEPTO / AÑOS	AÑO 01	AÑO 02	AÑO 03	AÑO 04	AÑO 05	
I. INGRESOS (S/.)						
1.-Total Ingreso	S/ 980.000,00	S/ 980.000,00	S/ 980.000,00	S/ 980.000,00	S/ 980.000,00	
Ingreso Presupuestal para El Seguro Agrícola Catastrófico	S/ 980.000,00	S/ 980.000,00	S/ 980.000,00	S/ 980.000,00	S/ 980.000,00	
II. EGRESOS (S/.)						
2.-Total Egresos	S/ 889.600,00	S/ 885.800,00	S/ 894.550,00	S/ 887.400,00	S/ 896.950,00	
Gastos Administrativos	S/ 132.500,00	S/ 130.500,00	S/ 135.000,00	S/ 131.300,00	S/ 136.000,00	
Gastos Logísticos	S/ 755.600,00	S/ 753.600,00	S/ 758.100,00	S/ 754.400,00	S/ 759.100,00	
Gastos de Suministros y servicios	S/ 1.500,00	S/ 1.700,00	S/ 1.450,00	S/ 1.700,00	S/ 1.850,00	
Utilidad Operativa	S/ 90.400,00	S/ 94.200,00	S/ 85.450,00	S/ 92.600,00	S/ 83.050,00	
Utilidad antes de Impuestos	S/ 90.400,00	S/ 94.200,00	S/ 85.450,00	S/ 92.600,00	S/ 83.050,00	
Impuesto a la Renta	S/ 13.560,00	S/ 14.130,00	S/ 12.817,50	S/ 13.890,00	S/ 12.457,50	
FCE ACTUAL	S/ 76.840,00	S/ 80.070,00	S/ 72.632,50	S/ 78.710,00	S/ 70.592,50	

Tabla 62: FLUJO DE CAJA ACTUAL – AGROEVALUACIONES

Fuente: Creación propia

2. FLUJO DE CAJA ECONÓMICO (INCLUYENDO LA IMPLEMENTACIÓN DEL SISTEMA AGRÍCOLA CATASTRÓFICO):

Como se explica en la tabla 63, ese es el flujo de caja económico proyectado de Agroevaluaciones Perú.

Tasa de Crecimiento:

0,180

FLUJO DE CAJA ECONÓMICO (INCLUYENDO LA IMPLEMENTACIÓN DEL SISTEMA AGRÍCOLA CATASTRÓFICO)						
CONCEPTO / AÑOS	AÑO 0	AÑO 01	AÑO 02	AÑO 03	AÑO 04	AÑO 05
I. INGRESOS (S/.)						
1.-Total Ingreso	S/980.000,00	S/ 980.000,00	S/ 980.000,00	S/980.000,00	S/ 980.000,00	S/ 980.000,00
Ingreso Presupuestal para El Seguro Agrícola Catastrófico	S/980.000,00	S/ 980.000,00	S/ 980.000,00	S/980.000,00	S/ 980.000,00	S/ 980.000,00
II. EGRESOS (S/.)						
Costo de Inversión						
Desarrollo del Sistema Agrícola Catastrófico + Mantenimiento						
IMPLEMENTACIÓN (TESISTA)	S/ 40.000,00	0	0	0	0	0
Mantenimiento Mensual (incluye Pago de Hosting, Dominio, Servidor Dedicado y Exclusivo, Base de Datos, control de incidencias, seguimiento de consumo de recursos y control de infraestructura), incluyen cambios	S/5.000,00	S/ 5.000,00	S/ 5.000,00	S/ 5.000,00	S/ 5.000,00	S/ 5.000,00
(Total de Inversión)	S/45.000,00	S/ 5.000,00	S/ 5.000,00	S/ 5.000,00	S/ 5.000,00	S/ 5.000,00
2.-Total Egresos	S/490.450,00	S/ 402.169,00	S/ 329.778,58	S/ 270.418,44	S/ 221.743,12	S/ 181.829,36
Gastos Administrativos	S/ 128.000,00	S/ 104.960,00	S/ 86.067,20	S/ 70.575,10	S/ 57.871,59	S/ 47.454,70
Gastos Logísticos	S/ 361.250,00	S/ 296.225,00	S/ 242.904,50	S/ 199.181,69	S/ 163.328,99	S/ 133.929,77
Gastos de Suministros y servicios	S/1.200,00	S/ 984,00	S/ 806,88	S/ 661,64	S/ 542,55	S/ 444,89
Utilidad Operativa	S/ 444.550,00	S/ 572.831,00	S/ 645.221,42	S/ 704.581,56	S/ 753.256,88	S/ 793.170,64
Utilidad antes de Impuestos	S/444.550,00	S/572.831,00	S/ 645.221,42	S/ 704.581,56	S/ 753.256,88	S/ 793.170,64
Impuesto a la Renta	S/ 111.137,50	S/ 143.207,75	S/ 161.305,36	S/ 176.145,39	S/ 188.314,22	S/ 198.292,66
FCE	S/ 333.412,50	S/ 429.623,25	S/ 483.916,07	S/ 528.436,17	S/ 564.942,66	S/ 594.877,98

Tabla 63: Flujo de caja proyectado – agroevaluaciones
Fuente: Creación propia

3. RELACIÓN COSTO – BENEFICIO, VAN y TIR:

En la tabla 64 podemos observar los análisis respectivos que se han realizado para encontrar el cálculo de las variables VAN, TIR y Costo beneficio; las cuales se trabajarán con una tasa de descuento del 10%.

CALCULO DEL VAN, R B/C Y TIR CON UNA TASA DE DESCUENTO DEL 10%

Año de operación	Costos totales (\$)	Beneficios totales (BT) (\$)	Factor de actualización (FA) 10,0%	Costos actualizados (CA) (\$)	Beneficios actualizados (BA) (\$)	Flujo neto de efectivo act. (\$)
(n)	(CT)	(BT)	$1 / (1+10\%)^n$	FA*CT	FA*BT	BA-CA
0	S/ -	S/ -	S/ 1,00	S/ -	S/ -	S/ -
1	S/ 535.450,00	S/ 333.412,50	S/ 0,91	S/ 486.772,73	S/ 303.102,27	-S/ 183.670,45
2	S/ 407.169,00	S/ 429.623,25	S/ 0,83	S/ 336.503,31	S/ 355.060,54	S/ 18.557,23
3	S/ 334.778,58	S/ 483.916,07	S/ 0,75	S/ 251.524,10	S/ 363.573,30	S/ 112.049,20
4	S/ 275.418,44	S/ 528.436,17	S/ 0,68	S/ 188.114,50	S/ 360.929,02	S/ 172.814,52
5	S/ 226.743,12	S/ 564.942,66	S/ 0,62	S/ 140.789,64	S/ 350.784,95	S/ 209.995,31
6	S/ 186.829,36	S/ 594.877,98	S/ 0,56	S/ 105.460,30	S/ 335.793,11	S/ 230.332,81
Total	S/ 1.966.388,49	S/ 2.935.208,63		S/ 1.509.164,57	S/ 2.069.243,19	S/ 560.078,62

Tabla 64: Relación Costo- Beneficio, VAN y TIR
Fuente: Creación propia

4. RESUMEN DE EVALUACIÓN ECONÓMICA:

Considerando que los valores obtenidos están dentro de las condiciones restrictivos podemos decir lo siguiente:

- ✚ El VAN tiene la siguiente condición: $VAN > 0$; "Se acepta"; "Se rechaza"
 - El VAN para nuestro caso es el valor final de la suma del flujo neto de efectivo actual.
- ✚ El TIR tiene la siguiente condición: $TIR > 10\%$; "Se acepta"; "Se rechaza"
 - El TIR para nuestro caso es la comparación de matrices de los costos y los beneficios totales.
- ✚ El B/C tiene la siguiente condición: $B/C \geq 1$; "Se acepta"; "Se rechaza"
 - El Costo – Beneficio es la división de los beneficios actualizados entre los costos actualizados.
 - Los costos actualizados lo obtenemos de la multiplicación del costo total por el factor de actualización.
 - Los beneficios actualizados lo obtenemos de la multiplicación del beneficio total por el factor de actualización.

Los indicadores financieros que arroja el proyecto los podemos apreciar en la tabla n° 65 son los siguiente:

VAN=	560.078,62
TIR =	65,43%
B/C =	1,37

Tabla 65: Resumen de evaluación económica - VAN-TIR-B/C
Fuente: Creación propia

5. FACTIBILIDAD DE MERCADO:

Se puede observar en la tabla 65, que el valor del Beneficio – Costo es mayor a 1 (S/ 1,37), podemos deducir que la implementación del Sistema de gestión, análisis y cartografía para el seguro agrícola catastrófico Lima – 2017 es factible, representando así un beneficio de S/ 560.078,62. La inversión realizada que asciende al monto de S/. 45 000.00 será retornada en el periodo de un año.

III. Anexo 3: Cronograma de Actividades:

Como se muestra en la imagen 126, el desarrollo de este sistema duró 19 meses (1 año 7 meses).



Imagen 126: Cronograma de Actividades
Fuente: Creación propia